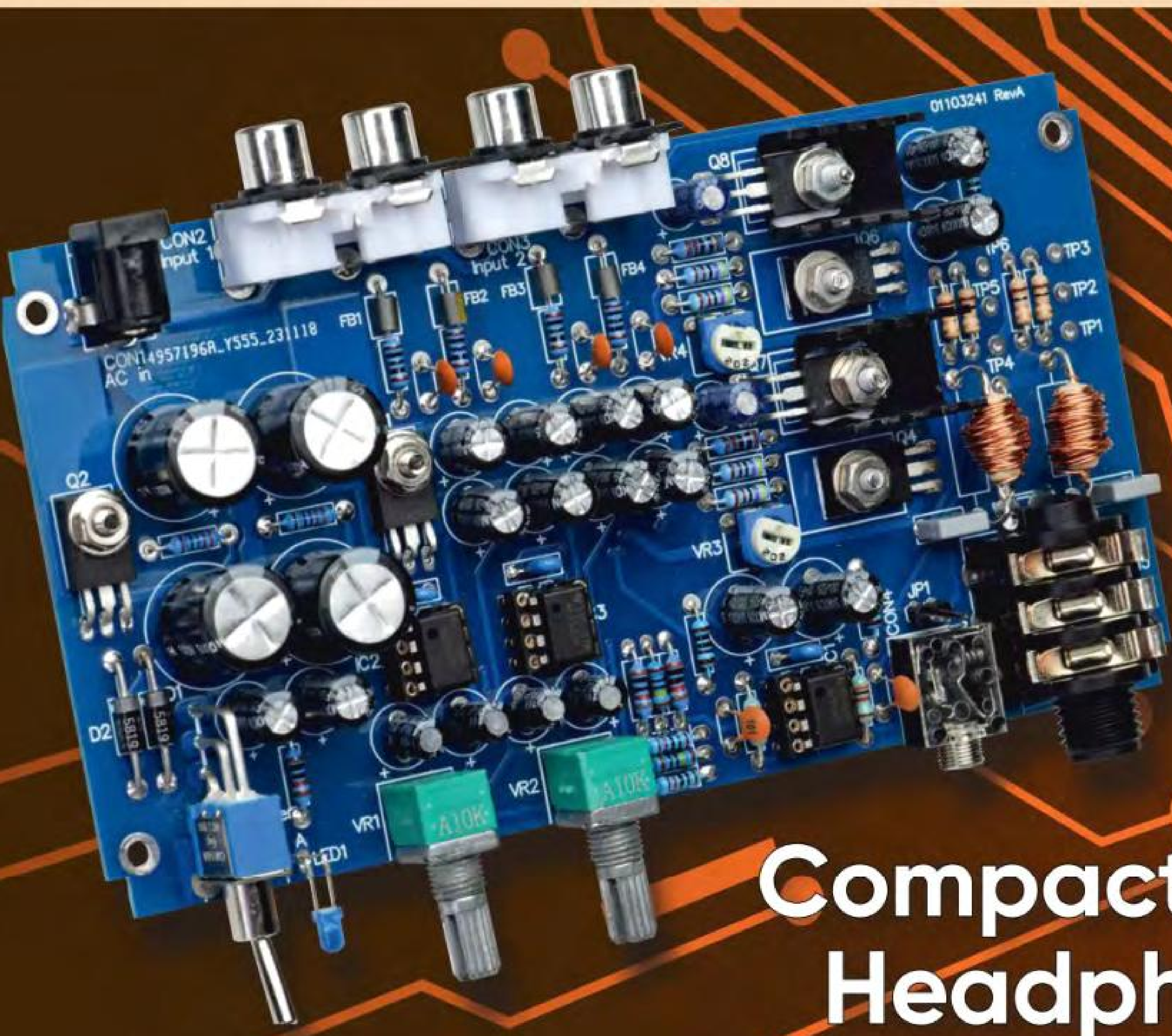
# Practical Electronics

The UK's premier electronics, computing and maker magazine

**Circuit Surgery**
Getting an FIR filter algorithm running on a microcontroller

**Audio Out**
A proper PCB for building the cascode audio amplifier

**Net Work**
Smart meters, new Chinese EVs & Temu



## Compact Hi-Fi Headphone Amp

## Discrete Ideal Bridge Rectifiers

LQ METER
Q=236 @ 6.28MHz
L=5.9uH C=114pF
L
SET HIGH
START
ON
SET C

## Automatic L/Q Meter

**PLUS!**

Techno Talk – the history of breadboarding
Cool Beans – a crash course in digital logic and gates
Dual Rail Supply Protector – a bipolar electronic fuse

Sep 2025 £5.99

www.electronpublishing.com | @practicalelec | practicalelectronics

PCBWay

YOUR SINGLE SOURCE FOR CIRCUIT EXCELLENCE

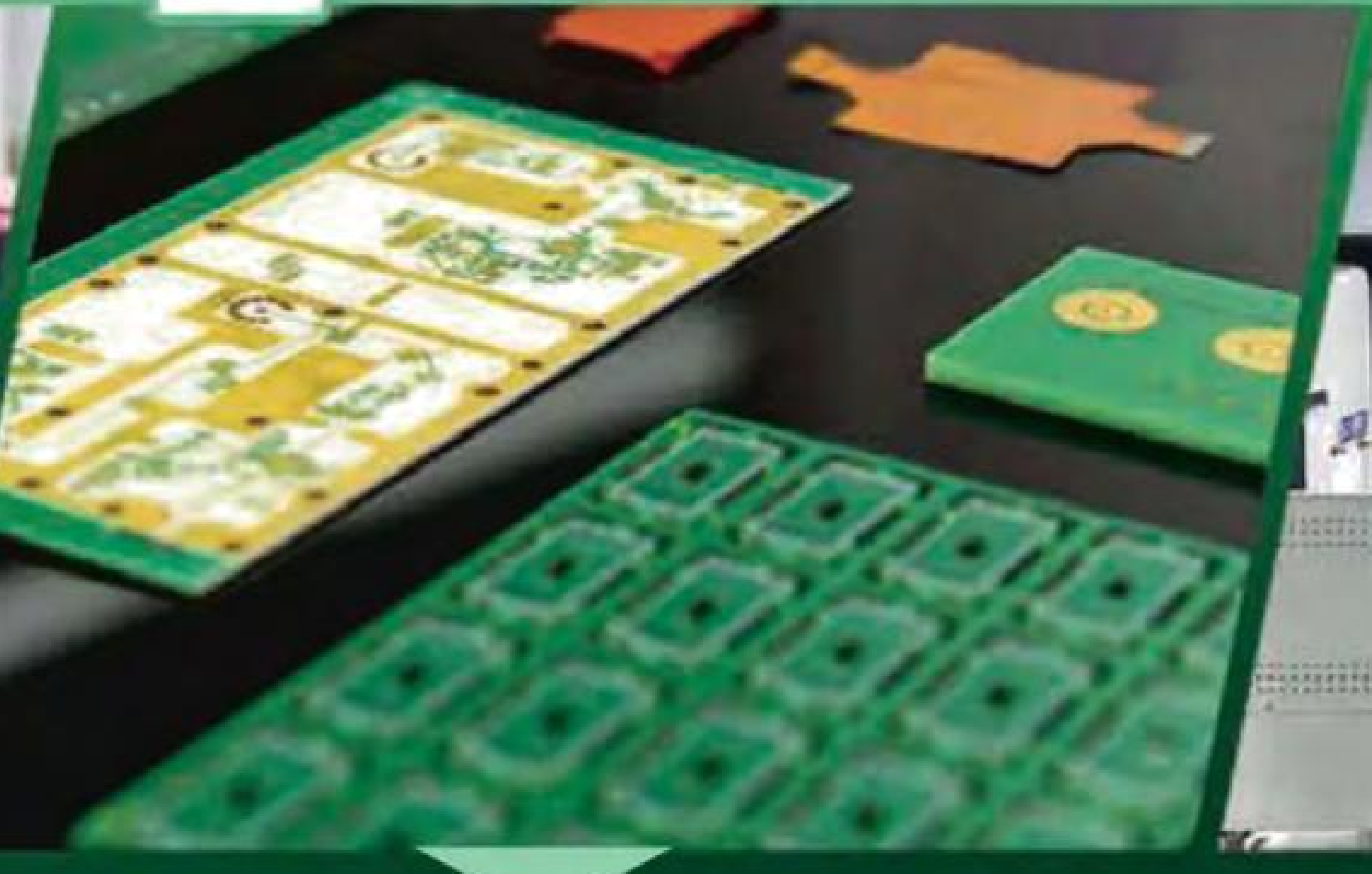
From PCB

To Assembly

Why choose PCBWay as your trusted partner?

1 One-Stop Solution

2 Broad Expertise



3 Premium Quality

From PCB design to PCBA, we offer comprehensive services.

Flex, HDI and aluminum PCBs are all within our capabilities.

Backed by CE, UL, ISO, REACH, RoHS certifications, etc.

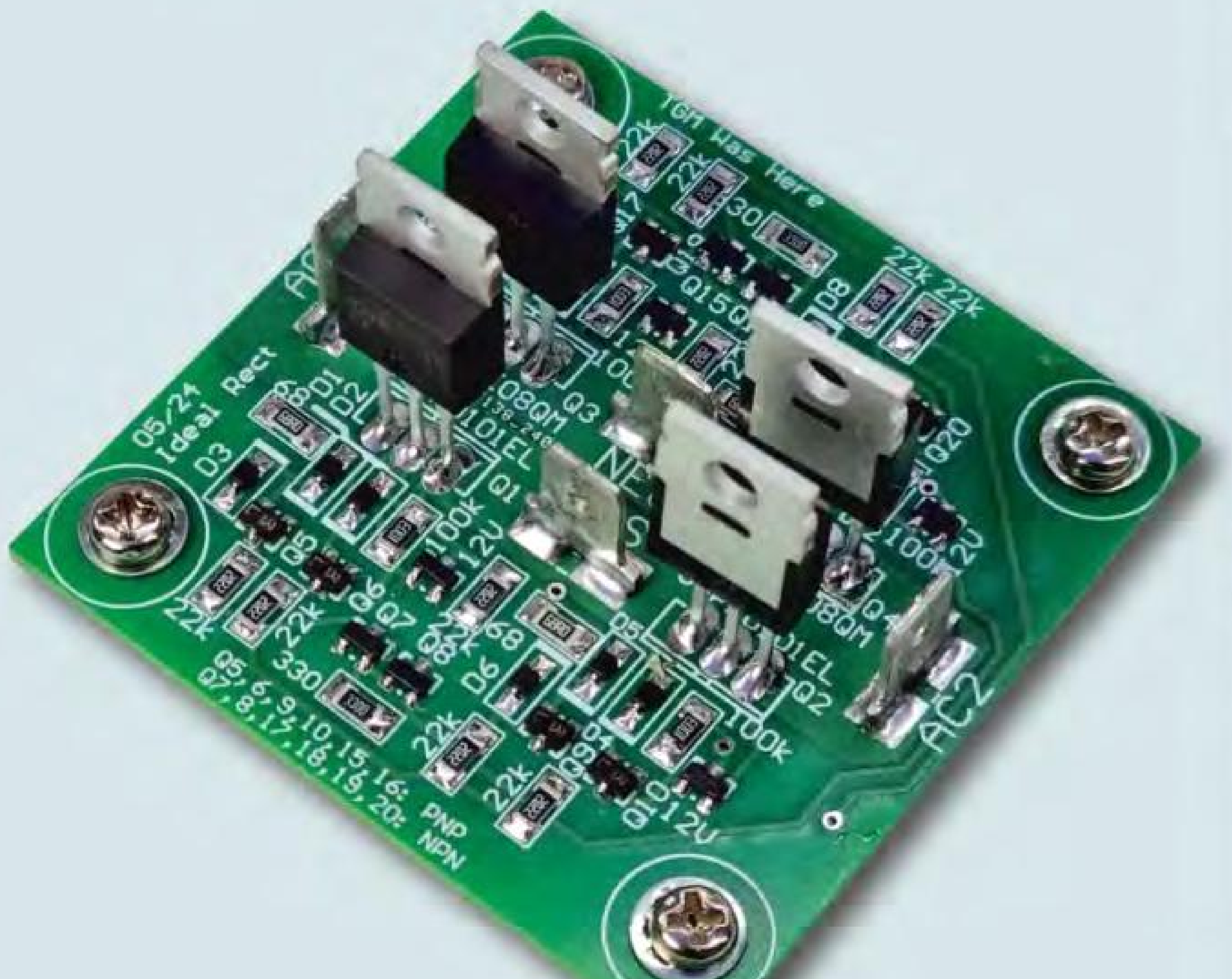
www.pcbway.com

service@pcbway.com

# Practical Electronics

Volume 54. No. 9
September 2025
ISSN 2632 573X

# Contents



**Made in Australia & the UK.**

**Written in Britain, Australia, the USA and Ireland.**

**Read everywhere.**

© Silicon Chip Publications Pty Ltd 2025

Copyright in all drawings, photographs, articles, technical designs, software and intellectual property published in *Practical Electronics* is fully protected, and reproduction or imitation in whole or in part are expressly forbidden.

The October 2025 issue of *Practical Electronics* will be published on Tuesday, 4th of September 2025 (see p80).

## Projects and Circuits

**Compact Hi-Fi Headphone Amplifier, part one** by Nicholas Vinen 4
This Headphone Amplifier is easy to build, sounds great, is inexpensive and fits into a compact case. It's ideal for beginners or those who want to get the best out of a set of traditional wired headphones. It's plugpack-powered, so no mains wiring is required.

**Two Discrete Ideal Bridge Rectifiers** by Phil Prosser & Ian Ashford 18
This simple circuit uses just a handful of transistors, diodes and resistors. But it still provides a very useful function: active rectification of the output of a centre-tapped transformer or combining two DC supplies with low losses.

**Automatic LQ Meter** by Charles Kosina 34
A Q Meter is an indispensable tool for anyone contemplating RF design. This new design features two instruments in one, measuring inductance from 0.1 to 999μH and Q from 10 to 300 with a test frequency from 100kHz to 90MHz!

**Dual-Rail Load Protector** by Stefan Keller-Tuberg 60
It disconnects a load from its power supply if the voltage is reversed or too high. It also disconnects the load if it draws current above the adjustable trip level. Its dual-rail support means it can work with devices like audio amplifiers with a split supply.

## Series, Features and Columns

**The Fox Report** by Barry Fox 3
A reasonably priced solution for high-quality analog audio to digital conversion.

**0.91-inch monochrome OLED display modules** by Jim Rowe 12
Small monochrome OLED display modules have become widely available at a low cost in the last few years. This one measures just 37.5 × 11.5 × 4.5mm but has a useful 128 × 32 pixel display that's either white or blue.

**Techno Talk** by Max The Magnificent 16
We take breadboards for granted, but the modern version evolved over several decades.

**Max's Cool Beans** by Max The Magnificent 26
Weird & Wonderful Arduino Projects, part 9: a crash course in digital logic.

**Net Work** by Alan Winstanley 44
The return of my occasional column brings news of trends in smart meters, the onslaught of Chinese-made electric vehicles, and tempts readers into trying Temu.

**Circuit Surgery** by Ian Bell 48
Topics in digital signal processing – Implementing DSP on a microcontroller, part 4. It is finally time to implement an FIR low-pass filter on our microcontroller board.

**Altium Designer 2025 review** by Tim Blythman 56
We look at the features and improvements added to this EDA software in its last update.

**Audio Out** by Jake Rothman 70
You can now easily build my cascode amplifier using this custom-designed PCB.

## Regulars and Services

**Editorial** 2
Values and units shouldn't be separated
**Reader services – Editorial and Advertising Departments** 2
**Subscribe to Practical Electronics and save money** 43
**Practical Electronics Back Issue DOWNLOADS – 2024 now available!** 69
**Practical Electronics PCB Service** 76
PCBs for Practical Electronics projects
**Classified ads and Advertiser index** 79
**Next month! – highlights of our next issue of Practical Electronics** 80

Renewals paid by PayPal, Visa and Mastercard are now handled via our website: **www.electronpublishing.com**

Practical Electronics

Volume 54. No. 9
September 2025
ISSN 2632 573X

# Editorial

**Editorial offices**
Electron Publishing (Australia)
**Tel** +61 2 9939 3295
**Email** pe@pemag.au
**Web** www.electronpublishing.com

**Address mail to:**
Electron Publishing (Australia)
PO Box 194, Matraville NSW 2036
Australia

**Advertising enquiries**
+61 2 9939 3295 pe@pemag.au

| | |
|---|---|
| **Editor** | Nicholas Vinen |
| **Publisher** | Nicholas Vinen |
| **Digital subscriptions** | Stewart Kearn **Tel 07918 614662** |
| **Online Editor** | Alan Winstanley |
| **Web Systems** | Kris Thain |
| **Production** | Bao Smith |
| **Technical staff** | Tim Blythman, John Clarke |

**Print subscriptions**
Practical Electronics Subscriptions
PO Box 6337
Bournemouth BH1 9EH
United Kingdom
**Tel** 01202 087631
**Email** pesubs@selectps.com

**Technical enquiries**
We regret that technical enquiries cannot be answered over the telephone. We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine.

Questions about articles or projects should be sent to the editor by email: pe@pemag.au

**Projects and circuits**
All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it.

Some projects and circuits published in *Practical Electronics* employ voltages that can be lethal. Do not build, test, modify or fix any mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD (GFCI) adaptor.

**Component supplies**
Silicon Chip Publications may offer kits or other parts for making our projects, but not in all cases. When kits are not available, readers will need to find and source parts themselves. We advise readers to check that all parts are still available before commencing any project in a back-dated issue.

**Advertisements**
Although the proprietors and staff of *Practical Electronics* take reasonable precautions to protect the interests of readers by ensuring as far as practicable that advertisements are bona fide, the magazine and its publishers cannot give any undertakings in respect of statements or claims made by advertisers, whether these advertisements are printed as part of the magazine, or in inserts. The Publishers regret that under no circumstances will the magazine accept liability for non-receipt of goods ordered, or for late delivery, or for faults in manufacture.

**Transmitters/bugs/telephone equipment**
We advise readers that certain items of radio transmitting and telephone equipment which may be advertised in our pages cannot be legally used in the UK. Readers should check the law before buying any transmitting or telephone equipment, as a fine, confiscation of equipment and/or imprisonment can result from illegal use or ownership. The laws vary from country to country; readers should check local laws.

## Why I don't put a space between numbers and units

International standard ISO/IEC 80000 states that "A space should be used to separate a numerical value and its unit symbol. This space is a non-breaking space and ideally a thin space in typesetting."

Reading *Practical Electronics*, you will have have noticed that we don't do that; we put units immediately after a number, for example, 12V, 1.3A or 25°C. This is not because we are lazy; it is a deliberate convention.

The reason for this is that, when a number has a unit attached, the numbers and units are inextricably linked. The units can only be changed by a mathematical transformation between that number/unit pair and another number/unit pair. To separate the numerical value any other way is to invite disaster.

So, for example, when calculating 12V × 10, the result is 120V, **not** 120. However, if you perform the calculation 12V ÷ 10V, the result is 1.2 – a unitless number, **not** 1.2V. In other words, when operating on either the number or the unit, you always operate on both.

I don't think that adding a thin space makes it look any better (you might disagree). But that is not why we don't do so; it's because we want to reinforce the idea that the two parts are inseparable.

This has real consequences. Too many students (and some professionals) treat units as afterthoughts, rather than integral parts of physical quantities. We believe that presenting values and units as a unified whole helps reinforce the correct mental model: that "12V" is not just a number with a label, but a distinct and meaningful quantity that cannot be casually separated.

There's also a growing awareness in fields like programming, data science and engineering software that quantities with units are fundamentally different types to plain numbers. Software libraries like Pint (Python) or units-aware spreadsheets make this distinction explicit. If you try to add 12V to 5A, you'll get an error – and rightly so. Keeping the unit visually connected to the number supports this idea of type safety and helps readers avoid conceptual errors.

Finally, it's worth acknowledging that typographic conventions were created in the print era, when spacing was as much about line justification and aesthetics as semantics. Today, we write and read in digital formats where cut-and-paste, machine parsing, and semantic clarity often matter more. A thin space may look elegant in typeset copy, but it makes it far easier for readers to accidentally copy just the number without the unit.

**Nicholas Vinen, Electron Publishing (Australia)***
**Publisher & Editor,**
**Practical Electronics Magazine**
* a division of Silicon Chip Publications Pty Ltd.

# The **Fox** Report

Barry Fox's technology column

## Analog2Digital

**I wanted an easy way to take analog audio** from a Hi-Fi system and turn it into digital data, in high quality, for easy storage.

The first challenge was how to get line audio out of an amplifier. Using the signal from the speakers or a headphone socket creates signal level problems.

Older Hi-Fi systems have stereo (left/right) fixed-level line output sockets, with phono/RCA plugs, for connecting a tape recorder or CD burner. However, these sockets are now seldom provided.

Often, the only option now is to use the line level outputs provided for connecting a separate amplifier or multi-room, multi-zone system. If there are multiple RCA/phono sockets, for surround sound and a woofer, use the front stereo pair. The system will probably need to be set up to steer a fixed-level replica of the main Zone 1 audio to the Zone 2 sockets.

Details will vary from amplifier to amplifier, and there is no alternative to reading the manual, Googling or asking an AI to figure out how to do this.

The next challenge is how to convert the analog, stereo line-out audio accurately into the digital domain.

There are many accessory boxes and dongles that can do the job. Cheap ones, often designed to digitise low-quality video and audio from a VHS VCR, may deliver iffy audio quality. Units that use high-quality ADCs (analog-to-digital converters) are expensive.

When I recently looked for something affordable to use for digitising old audio tapes, discs or radio broadcasts, with halfway-to-decent audio quality, I found a big gap in the market.

Modern LP turntables now often have a built-in ADC and USB socket that provides a digital version of the analog vinyl music. But this is no help when trying to digitise other sources.

Highly respected Austrian Hi-Fi company Pro-Ject has for many years made a small accessory box, costing around £200. This takes in the very low-level "phono" feed from MC (moving coil) or MM (moving magnet) turntable cartridges and uses a pre-amp, equaliser and high-quality ADCs to produce a line-level signal for connection to an amplifier without a phono input, plus a digitised stream via USB.

I suggested to Pro-Ject that it might be a smart move to add line level inputs; that way, the expensive ADCs could be used for two tasks, digitally converting from phono level signals and also from line level signals. Surely all it would need, I suggested, would be a simple pre-amp/EQ bypass switch.

Scott Lawson, Technical Services Manager for Henley Audio, the British company that handles the Pro-Ject brand, was quick to shoot down my suggestion:

"Bypassing the 'pre-amp' doesn't make sense for a phono stage, because all you'd be left with is a very low signal output, which wouldn't be much use for recordings via the USB.... I doubt this is of any interest to Pro-Ject, as their market is predominantly turntables."

Fortunately, a clever new device from enterprising British company A2D2 (**https://a2d2.net/**) goes a long way towards doing the conversion job, plus a lot more besides, for just £150.

The A2D2 Stream is a small box that takes any line-level analog stereo signal (from disc, tape, radio etc) and converts it into a digital stream, which can then be shared by WiFi, Bluetooth or Ethernet cable to streaming players and amps, smart speakers, phones, tablets or computers.

The digital stream can be accessed by pointing a web browser at the device address (**http://a2d2.local/**). The browser then shows a list of the many open source and proprietary playback options that the A2D2 device supports.

These include AirPlay, ChromeCast, HEOS (Marantz/Denon), Sonos and Alexa (with some extra tweaking needed), standard Bluetooth or headphone pairing and direct "Listen on this Device" (eg, decode and play with a computer sound card) with the choice of 48kHz or 96kHz audio quality.

The device also has analog line output sockets so it can 'pass through' the signal to another analog audio device.

The software-based options are so wide and varied that I am still experimenting. For one practical test, I took the analog stereo Line Out from a Hi-Fi amplifier (a fixed-level Zone 2 connection for multi-room playback) and fed it into the A2D2 box for conversion into a digital stream. This then went by Ethernet cable into my home network.

To play the stream, I entered the LAN web address (**http://a2d2.local/**) in my desktop computer Chrome browser, chose "Play on this Device" and was offered the option to click "Play".

I could then listen to the stream through the computer's speakers or, more usefully, capture the stream, for instance with the open-source, free software Audacity. Just set the Audacity input to "Speakers".

To complete the experiment, I connected the A2D2 line out (pass-through) sockets to a stereo cable, which then ran to the Line In socket of the desktop computer. This provided an easy playback and capture choice between a) digital streaming via the home network and b) direct line level connection by analog stereo cable.

I am reserving judgement on whether the capture quality is better by network or line cable. What matters is that there is a simple, working choice available.

One practical point to note is, because the A2D2 box has an http web address (rather than https), your browser may show a security warning. This can be safely bypassed because the connection is local, not via the Wild West Internet.

The makers say that for streaming speakers, the A2D2 box self-adjusts quality to match the connected device; for example, 44.1kHz for Airplay, 48kHz for Sonos and 96kHz for Chromecast.

Device control is mainly with the (free) A2D2 mobile App.

I was disappointed to find no option to feed an analog signal into the A2D2 box and get out a USB digital stream. But this appears to be on a roadmap for the future, along with the opportunity for MP3 and FLAC encoding. In addition to the USB-C socket used for 5V DC power, there is already a standard USB output socket on the box, earmarked for data input and/or output.

It's not clear whether these and other future options, such as DLNA/UPnP support, NAS and Alexa integration, will be added by software updates. Hopefully so, but even if not, the A2D2 box is already a remarkably versatile bit of kit which, for £150, comes close to offering all things for all users.

The problem facing the A2D2 sales team may be that technically challenged influencers and casual gadget reviewers will not devote the time and effort needed to get to grips with the product.

I freely admit that I still feel I have only scratched the surface of its potential. But I have already understood more than enough to heartily recommend what could well become a standard tool for streaming.

PART 1: BY NICHOLAS VINEN

# COMPACT Hi-Fi HEADPHONE AMPLIFIER

This Headphone Amplifier is easy to build, sounds great, doesn't cost too much to make and fits into a compact instrument case. It's ideal for beginners or just those who want to get the best out of a set of traditional wired headphones. It's powered by a plugpack, so no mains wiring is required.

**It has been a while since we've published a headphone amplifier.** The reason I decided to design a new one is that the last one I designed, way back in 2011, had excellent audio quality, but was a bit overkill for many people. It was fairly large, somewhat expensive to build and consumed a fair bit of power, but you couldn't really fault the resulting sound quality, which was on par with some of the best power amplifiers.

Many of the headphone amplifier circuits I see published in various places are little more than an op amp, possibly with current-boosting transistors. Such designs work, but they have modest output power and usually less-than-stellar audio quality, although they may be considered adequate by many people.

I thought there was room for something in between: an amplifier with excellent audio quality that fit neatly into a compact case and wasn't too difficult or expensive to build.

That's precisely what this is. It's also beginner-friendly and has the handy feature of two stereo inputs that are mixed with independent volume controls.

That means you can connect two sound sources such as a TV and a computer, a CD player and a TV or something like that. With the separate volume controls, it's easy to account for different output levels from those devices, and you can also easily mute one if both are active. If you want to save time and money, you can build it with just one stereo input.

You have the choice of 3.5mm or 6.35mm jack sockets for the output (or both, optionally connected in parallel). Power is from a 9-12V AC 1-2A plugpack, a type that's readily

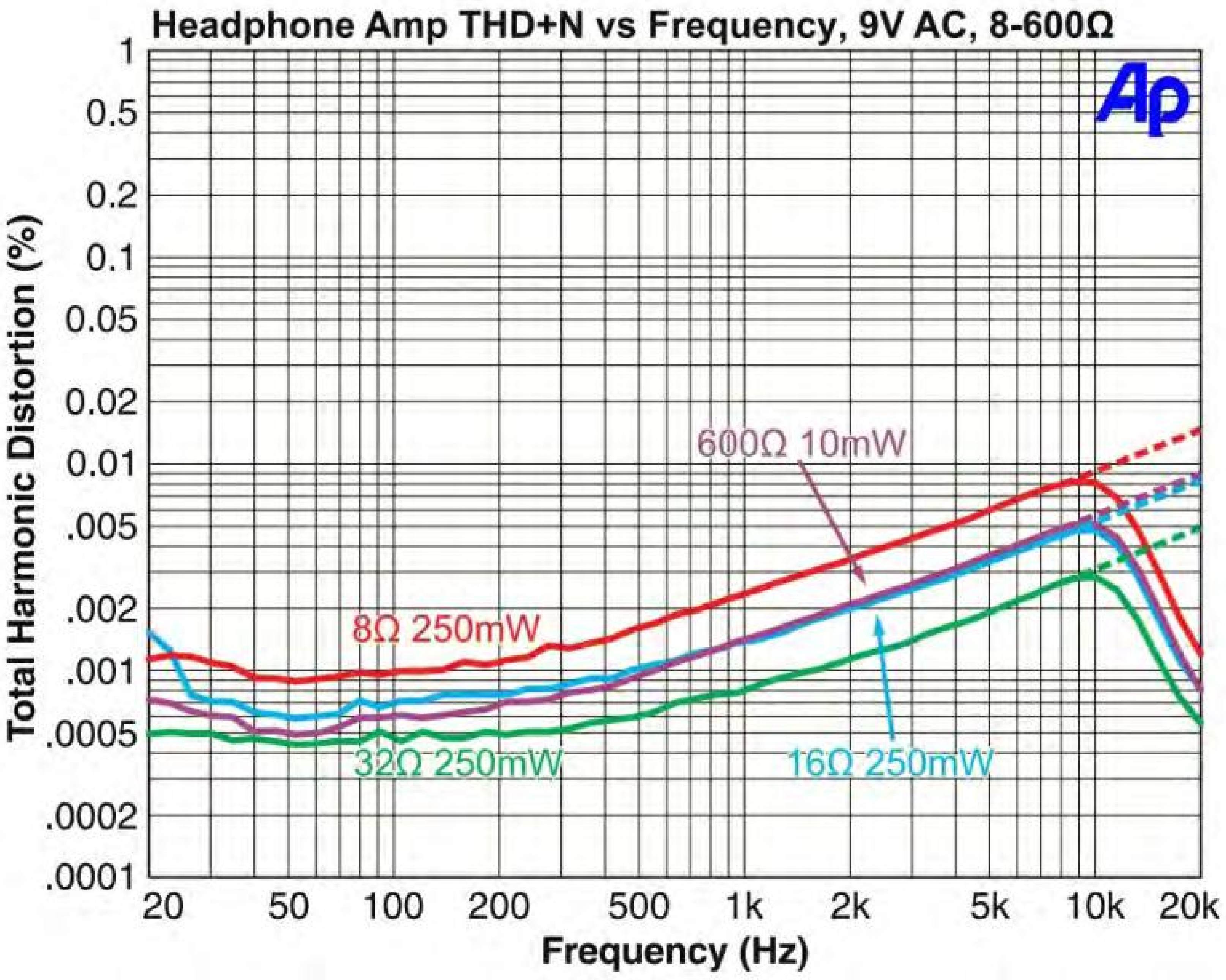


Fig.1: the Amp's distortion versus frequency for four common headphone/earphone load impedances. Distortion is lower for higher load impedances due to the lower output current required; the 600Ω curve is higher mainly due to the lower test power due to voltage swing limitations.

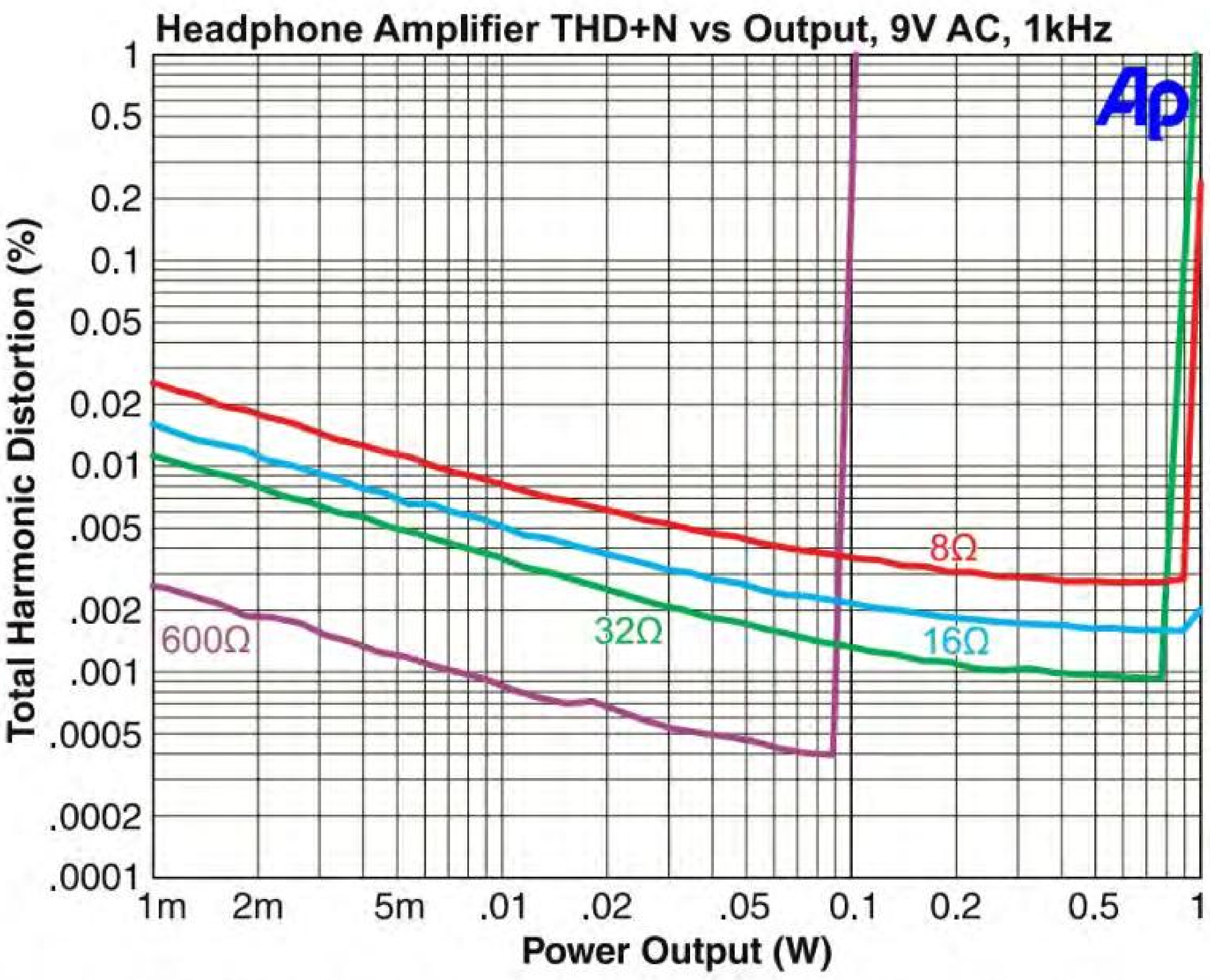


Fig.2: this shows how distortion varies with the output power level, at a fixed frequency. The onset of clipping is around 0.9W for an 8Ω load, due to current delivery limitations; a little over 1W for 16Ω; around 0.75W for 32Ω; or 90mW for a 600Ω load due to voltage swing limitations.

### Features & Specifications

- Drives stereo headphones with impedances from 8Ω and up
- Two outputs to suit 3.5mm or 6.35mm jack plugs
- Two stereo RCA inputs with independent volume controls
- Powered by a 9-12V AC plugpack
- Power on/off switch and power indicator LED
- Signal-to-noise ratio: 103dB with respect to 250mW into 8Ω
- Total harmonic distortion: <0.0025% @ 1kHz, <0.01% @ 10kHz (see Figs.1 & 2)
- Frequency response: 10Hz to 100kHz, +0,-0.2dB (16Ω load; see Fig.3)
- Channel separation: >70dB @ 1kHz (see Fig.4)
- Maximum output power (9V AC supply): 0.9W into 8Ω, 1W into 16Ω, 0.75W into 32Ω, 80-140mW (12V AC) into 600Ω
- Class-AB operating mode (Class-A at lower power levels)
- Inexpensive and easy to build
- Fits into compact 155×86×30mm ABS instrument case

**SILICON CHIP Kit (SC6885; ~£35)**
Includes the case but not a power supply

available from most suppliers. There is an onboard power switch and power indicator LED.

The headphone amplifier section is based on common low-noise, low-distortion op amps with transistor buffers to boost the output current. It will drive any headphones from 8Ω to 600Ω. It won't deliver a ton of power, but should be more than enough for any headphones, up to a watt (or maybe more) per channel.

If you really wanted to, you could use it to drive a pair of high-efficiency speakers to modest sound levels (eg, for use with a computer). While it isn't really designed for that task, it will work as long as the speakers are efficient enough and you're close to them.

This design uses all through-hole parts and it fits into a really nice little snap-together compact case that's just 155mm wide, 30mm tall and 86mm deep. So it takes up barely any room. The modest power consumption means it only gets a little warm during typical use, despite being unvented.

There's really nothing tricky to the construction. The only slightly fiddly bits are winding the inductors for the output filter (which only takes a few minutes) and mounting the output transistors and heatsinks, which is only difficult because the thermal paste can get on your fingers.

There is one adjustment per channel for quiescent current. It's easy to make by monitoring the voltage between pairs of test points with a DMM while twiddling a trimpot.

With a circuit that isn't too difficult to understand and straightforward construction, this should be a good project for relative beginners.

## Performance

At low signal levels, up to around 5mW (8Ω), 10mW (16Ω) or 20mW (32Ω/600Ω), the Headphone Amplifier operates in Class-A mode. Many headphones and earphones will produce reasonable volume levels at such powers. If your headphones require more power, or there are loud transients (like drum hits), the amplifier will automatically switch to Class-B (this is known as Class-AB operation).

The resulting performance is pretty good – not as good as our very best amplifiers, but certainly well above average. It's better than 'CD quality' under most conditions (which equates to about 0.0018% distortion at 1kHz with a 96dB signal-to-noise ratio).

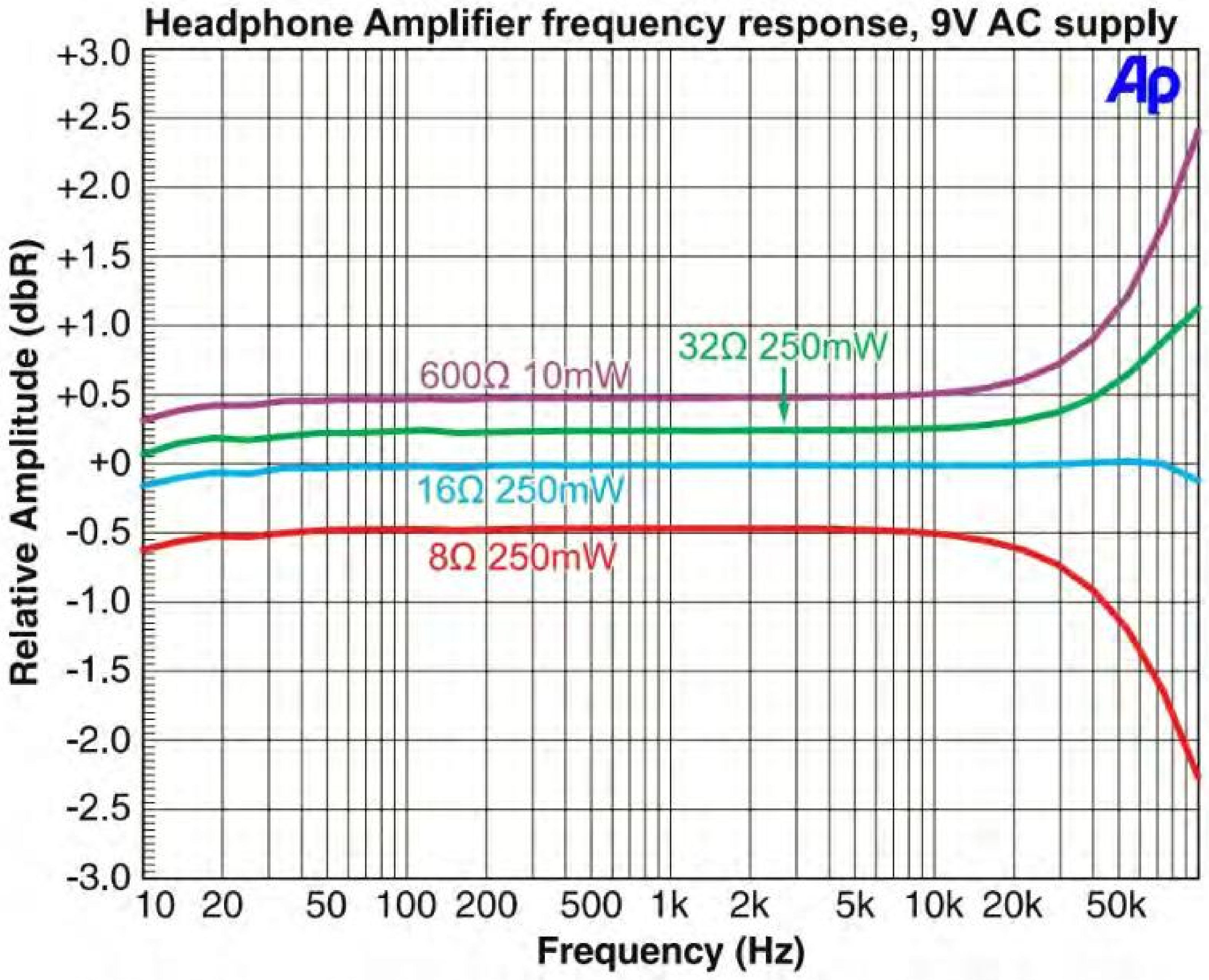


**Fig.3: the Amp's frequency response is very flat for all load impedances within the audible range (20Hz–20kHz). The deviation above 20kHz is due to the output filter. The vertical shifts are due to the Amp's output impedance (the level reduces slightly for lower load impedances).**

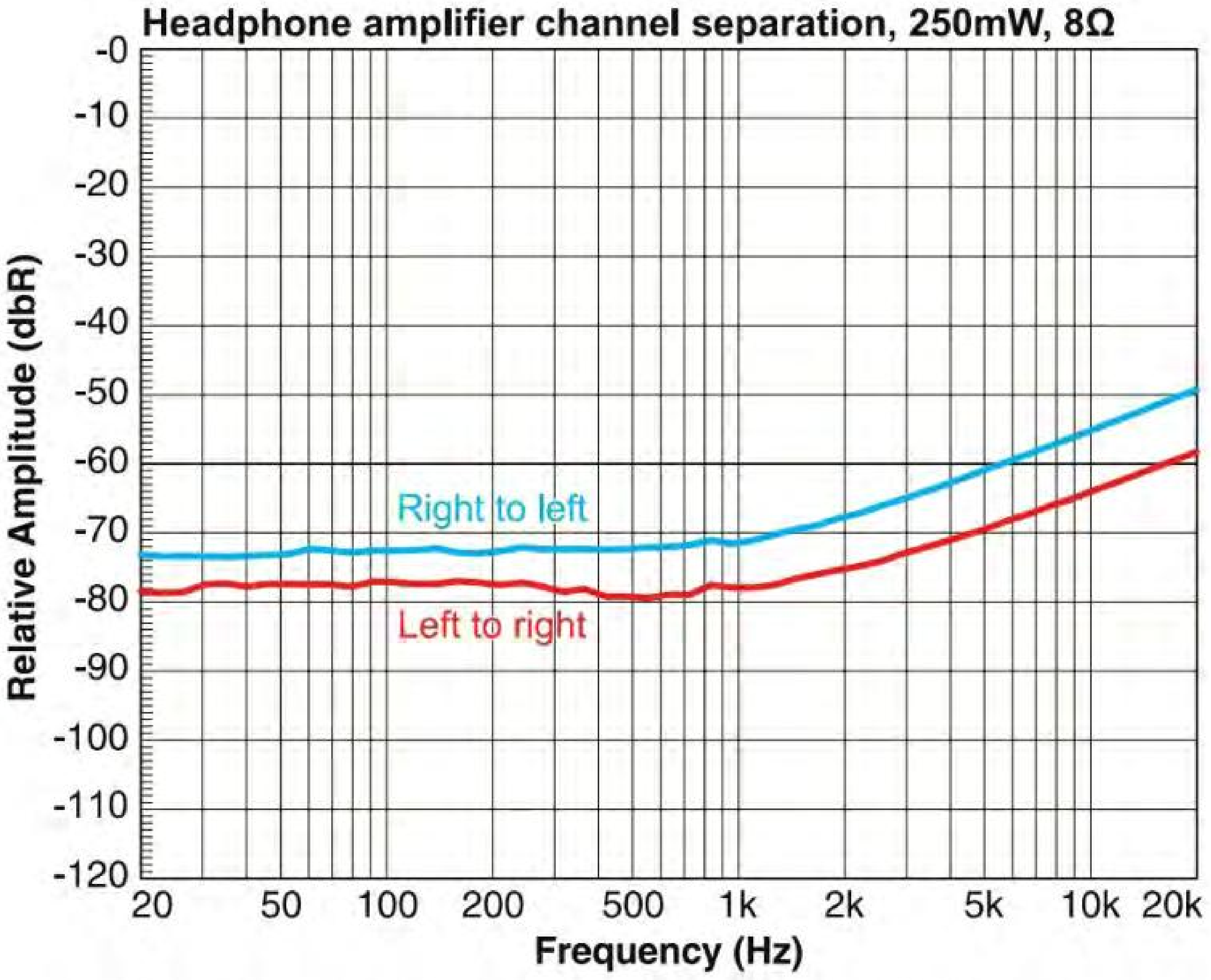
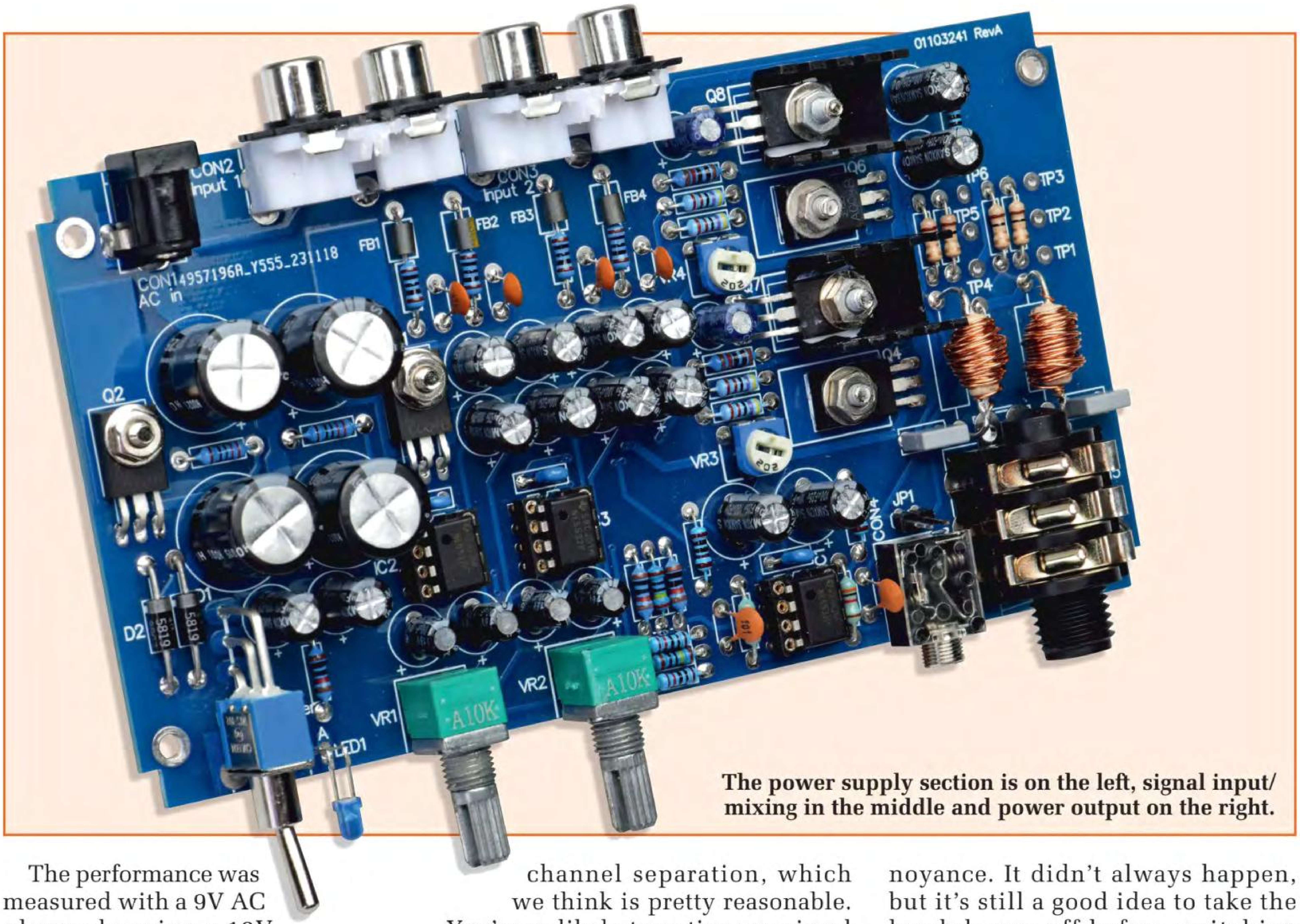


**Fig.4: there's a small amount of signal bleed between channels but it's attenuated by more than 70dB at 1kHz and below, so it is unlikely to be noticeable. Most stereo content has less separation than this anyway.**



The power supply section is on the left, signal input/mixing in the middle and power output on the right.

The performance was measured with a 9V AC plugpack; using a 12V plugpack will give the same or better performance.

Fig.1 shows how the total harmonic distortion plus noise (THD+N) level varies with frequency at 250mW (a high level for headphones!) into four common headphone load impedances. The performance is excellent for 32Ω headphones, well below 0.001% even up to several kilohertz. It's almost as good for 16Ω, reaching only around 0.0015% at 1kHz for 16Ω & 600Ω loads.

Even for the relatively low impedance of 8Ω, more typical for loudspeakers, the THD+N is just 0.0025% at 1kHz for a fairly high output level (250mW) and remains below 0.01% up to 10kHz.

Fig.2 shows how THD+N varies with power level. As the performance is essentially limited by noise, it is a steadily descending line until the point where it goes into clipping. That figure will give you a pretty good idea of how much power can be delivered with the 9V AC supply.

Fig.3 shows the frequency response, which is basically flat across the audible spectrum. Fig.4 shows the channel separation, which we think is pretty reasonable. You're unlikely to notice any signal bleeding between the channels.

Note that the maximum power delivery into high-impedance loads will depend on the supply voltage. Testing with a 9V AC plugpack, we got around 90mW into a 600Ω load before clipping, but we'd expect closer to 150mW with a 12V AC plugpack. Most headphones and earphones are well below 600Ω, so they are unlikely to run into voltage swing limitations even with a 9V AC supply.

## Subjective testing

I tested the Amp with a pair of Philips SHP9000 32Ω headphones (which, in my opinion, are excellent). As expected based on the flat frequency response and low distortion, the sound quality was top-notch, with lots of punchy bass, plenty of treble and no audible noise or artefacts.

There was no noticeable noise at switch-on with the headphones plugged in, although more sensitive headphones may make a noise. There was sometimes a modest crack or thump sound at switch-off, although it was not loud enough to cause anything more than annoyance. It didn't always happen, but it's still a good idea to take the headphones off before switching the amplifier off.

We also tested it by plugging in an Exteek C28 Bluetooth adaptor. We connected it to one input using a 3.5mm jack to twin RCA plug lead. That worked fine, and the Amp's gain was more than enough to drive the headphones to deafening levels from its relatively low-level output.

## Circuit details

The full circuit diagram is shown in Fig.5. We'll start by describing the input section and volume control, then the power amplification section, then the power supply. This description is for the full version of the circuit; later, we'll explain two ways it can be cut down.

The stereo input signals are applied to either of dual RCA sockets CON2 & CON3. They pass through an RF rejecting filter comprising ferrite beads, 100Ω series resistors and 470pF ceramic capacitors to ground. This should help eliminate any RF (eg, AM radio or switch-mode hash) picked up by the signal leads that could otherwise be demodulated by the following circuitry.

The signals are then AC-coupled using back-to-back polarised electrolytic capacitors. This is a cheaper and generally more compact configuration than non-polarised electrolytic capacitors, and has no real disadvantages. We use high-value coupling capacitors to retain good bass response, it also keeps the source impedance low for the following stages, to avoid noise creeping in.

The capacitor voltage ratings here are pretty high, so that if a faulty signal source delivering +18V or -18V DC (or more) is connected to one of the inputs, it won't damage anything.

It's important to AC-couple signals to potentiometers to avoid crackle when they are rotated. The signal is applied to the top of the potentiometers, which act as variable voltage dividers, the attenuated signal appearing at the wiper.

The potentiometers have a 'logarithmic taper', which is suitable for volume control since it better matches the way we hear loudness. Linear potentiometers tend to give poor control at the lower end of the volume range.

From the potentiometer wipers, the signals are again AC-coupled to the following op amp buffer stages, so that the op amp bias currents don't cause a DC voltage across the pots. Otherwise, that can also cause crackle when the pots are rotated.

Here we only need a polarised capacitor because we know the op amp input will be slightly positive due to the bias current flowing out of it. That is true for either of the op amp alternatives specified (NE5532 or LM833, which should both perform well). 100kΩ resistors to ground both DC-bias their input signal to 0V and provide a path for that bias current to flow.

The signals from the two pairs of buffers are then mixed using 10kΩ resistors and the mixed audio is fed to the power amplifier, on the right-hand side of the diagram. The 1MΩ resistors to ground provide a path for IC1's input bias currents to flow without IC2 and IC3 having to sink it, although the circuit would still work if those resistors were left out.

## Parts List – Compact Headphone Amplifier

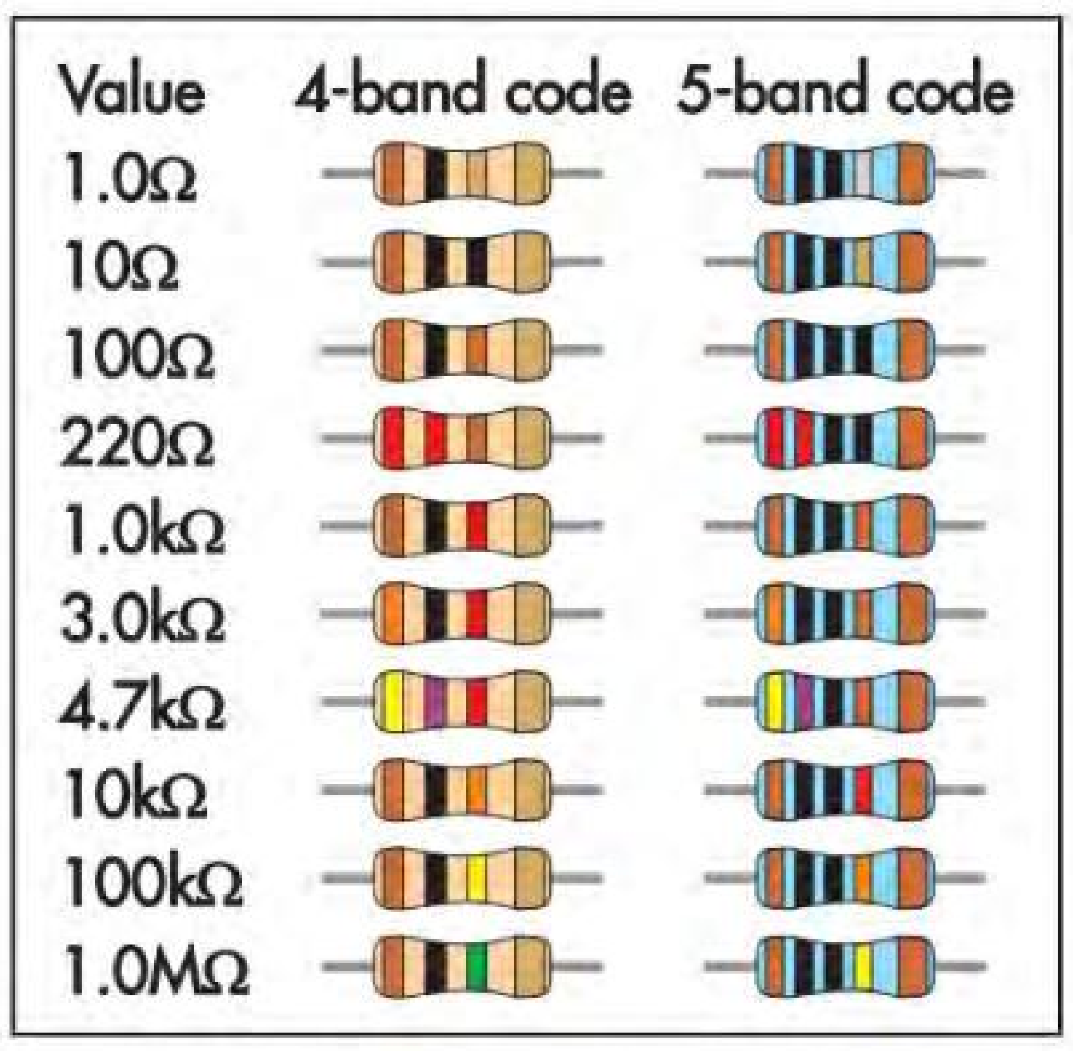
1 double-sided blue PCB coded 01103241, 148 × 80mm
1 Gainta G1816 or equivalent 155×86×30mm ABS instrument case
**[TME, Mouser 563-PC-11477]**
1 9-12V 1-2A AC plugpack
1 PCB-mount right-angle miniature SPDT toggle switch (S1)
1 PCB-mount barrel socket to suit plugpack (CON1)
2(1) dual horizontal white/red RCA sockets (CON2, CON3)
**[RCA-210; AliExpress 1005001629197904, Silicon Chip SC4850]**
1 PCB-mounting DPST 3.5mm stereo jack socket (CON4) ***AND/OR***
1 PCB-mounting low-profile DPST/DPDT 6.35mm stereo jack socket (CON5)
4(2) small ferrite beads (FB1-FB4)
1 2-pin header with jumper shunt (JP1)
2(1) 10kΩ dual-gang logarithmic taper 9mm right-angle PCB-mount potentiometers (VR1, VR2)
2 2kΩ top-adjust mini trimpots (VR3, VR4)
3(1) 8-pin DIL IC sockets (**optional**, for IC1-IC3)

**Wire & hardware**
1 2m length of 0.25-0.4mm diameter enamelled copper wire (for L1 & L2)
2 M3 × 16mm panhead machine screws
4 M3 × 10mm panhead machine screws
6 M3 flat washers
6 M3 hex nuts
4 No.4 × 5-6mm panhead self-tapping screws
2 TO-220 micro-U flag heatsinks (15 × 10 × 20mm)
2(1) small knobs to suit VR1 & VR2
4 small self-adhesive rubber feet

**Semiconductors**
3 NE5532 or LM833 low-noise, low-distortion op amps (IC1-IC3) ♦
5 TTC004B 160V 1.5A NPN transistors, TO-126 (Q1, Q3, Q5, Q7, Q8)
3 TTA004B 160V 1.5A PNP transistors, TO-126 (Q2, Q4, Q6)
1 3mm blue LED with diffused lens (LED1)
2 1N5819 40V 1A schottky diodes (D1, D2)
♦ *only one is required for cut-down version (unbuffered or single-channel)*

**Capacitors (maximum 20mm height)**
4 1000µF 25V low-ESR electrolytic (5mm pitch, maximum diameter 13mm)
2 470µF 10V electrolytic (5mm pitch, maximum diameter 10mm)
8(4) 100µF 50V electrolytic (5mm pitch, maximum diameter 8mm)
4 100µF 25V low-ESR electrolytic (5mm pitch, maximum diameter 8mm)
4(2) 100µF 16V electrolytic (5mm pitch, maximum diameter 8mm)
2 10µF 50V electrolytic (2.5mm pitch, maximum diameter 6.3mm)
2 100nF 63V MKT
3(1) 100nF 50V MKT, ceramic or multi-layer ceramic
4(2) 470pF 50V NP0/C0G ceramic
2 100pF 50V NP0/C0G ceramic

**Resistors** *(all ¼W 1% unless noted)*
2(0) 1MΩ
4(2) 100kΩ
7(3) 10kΩ
4 4.7kΩ
2 3kΩ
4 1kΩ
2 220Ω
4(2) 100Ω
2 10Ω 1W 5%
4 1Ω ½W (5% OK)


Value
4-band code
5-band code
1.0Ω
10Ω
100Ω
220Ω
1.0kΩ
3.0kΩ
4.7kΩ
10kΩ
100kΩ
1.0MΩ

**n** *number in bracket refers to quantities for the single-channel version*

## Power amplifier

This section is based on dual low-noise op amp IC1 and medium-power

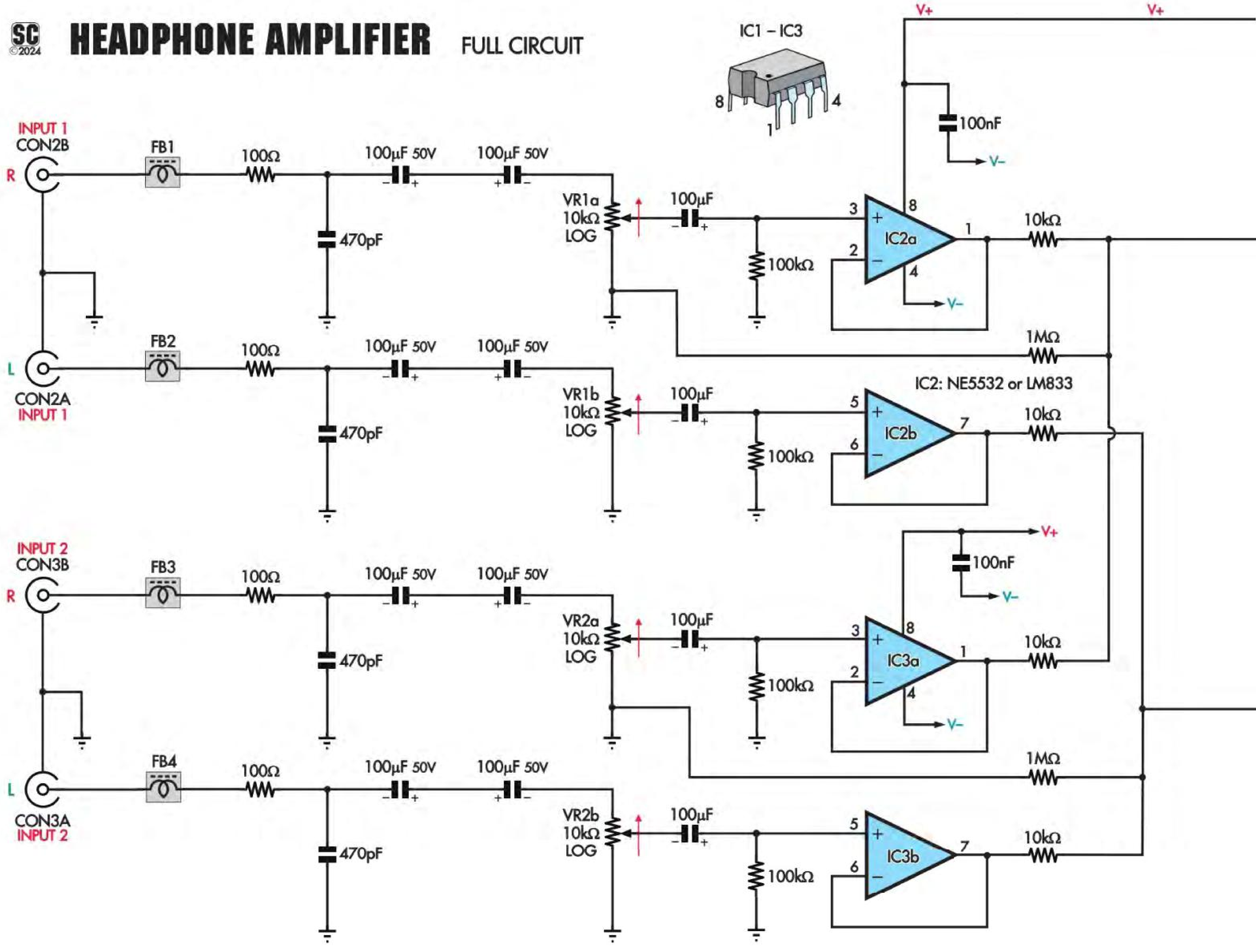


Fig.5: the full Headphone Amplifier circuit; the two stereo inputs are at upper left, the buffer and mixer left of centre, the output section at upper right and the power supply at lower right. It's all pretty conventional, but note the use of capacitance multipliers rather than regulators to provide reasonably steady V+ and V– rails without requiring a specific AC supply voltage.

transistors Q3-Q8. As the left and right channels are essentially identical, we'll stick to describing the right channel, with the corresponding left-channel designators being given in brackets (parentheses).

The incoming signal is fed into the non-inverting input, pin 3, of IC1a (IC2b). IC1a is configured as a non-inverting amplifier with a default gain of four times (12dB), although that can be changed by varying the 3kΩ and 1kΩ resistor values between the output and the feedback point, the pin 2 inverting input of IC1a.

The bottom end of the divider is connected to signal ground via a 470µF capacitor rather than directly, reducing the amplifier DC gain to unity. That way, the circuit doesn't amplify the op amp's inherent offset voltage (or any other offsets in the circuit).

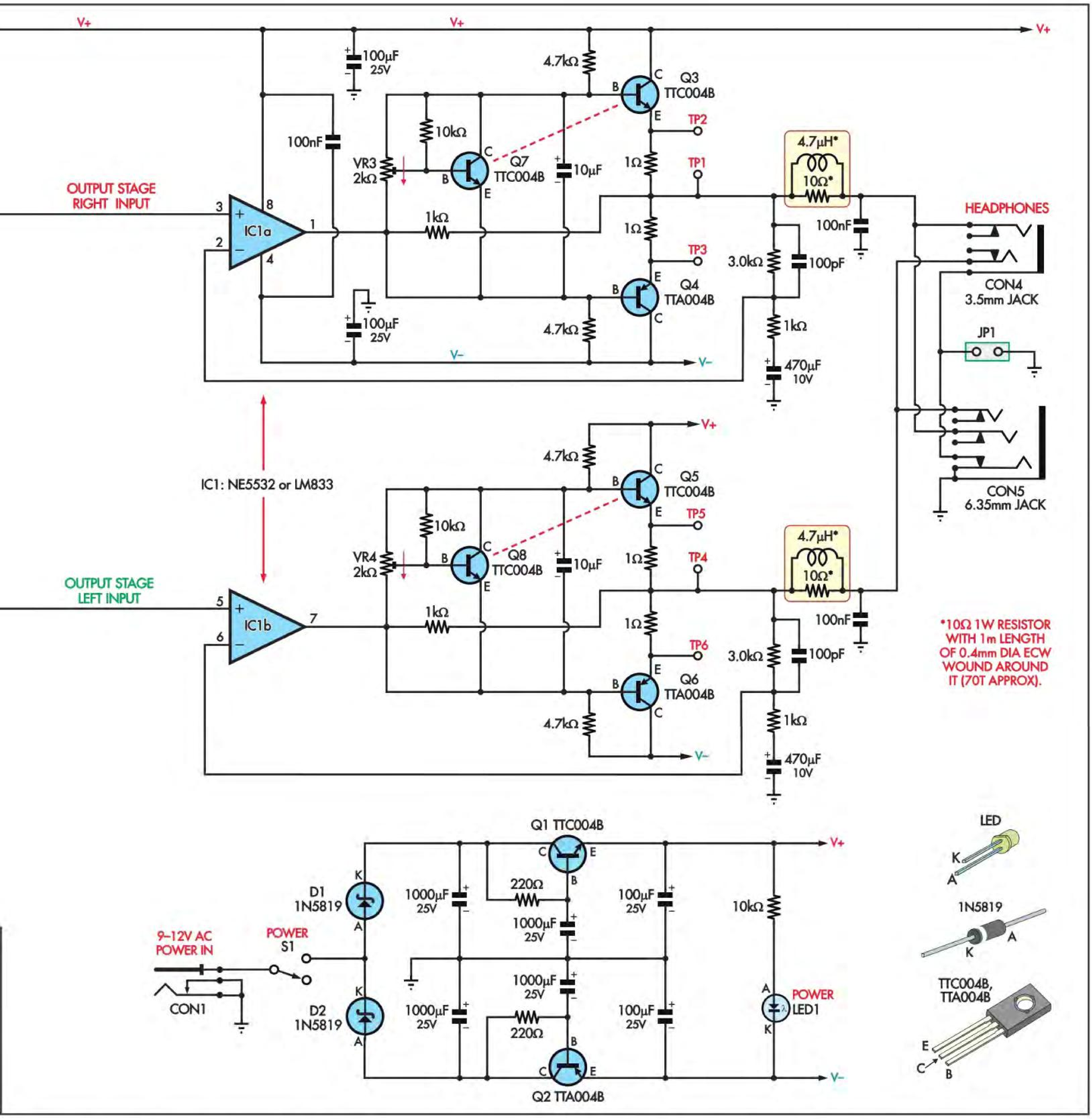
Most of the current to drive the headphones is supplied by NPN transistor Q3 (Q5) and PNP transistor Q4 (Q6), which are complementary emitter-followers. As the base voltage of Q3 rises, it sources more current into the output via its 1Ω emitter resistor, reducing its base-emitter voltage until it stabilises.

Similarly, when Q4's base is pulled down, its emitter pulls the output down and it too stabilises at a more-or-less fixed base-emitter voltage differential.

As Q3 and Q4 both have base-emitter voltage drops of around 0.7V when conducting a few milliamps, if we arrange for a difference of around 1.5V between the two bases (with Q3's base voltage being higher than Q4's), a small amount of current will constantly flow from the V+ rail, through Q3, the two 1Ω emitter resistors, then Q4 and back to the V- rail. This is called the quiescent current.

By having a small quiescent current, we keep Q3 and Q4 in conduction all the time, and we only have to vary the amount of conduction to smoothly control the output signal, rather than switching Q3 or Q4 on when needed. This is called Class-AB (sometimes Class-B) and it has the benefit of minimising (and ideally, virtually eliminating) crossover distortion.

Crossover distortion is an undesirable step in the output voltage as it passes through 0V, which an AC audio signal does frequently.


V+
V+
V+
100µF 25V
100nF
4.7kΩ
Q3 TTC004B
B
C
E
TP2
10kΩ
VR3 2kΩ
Q7 TTC004B
10µF
1Ω
TP1
4.7µH*
10Ω*
OUTPUT STAGE RIGHT INPUT
3
2
8
4
1
IC1a
1kΩ
1Ω
TP3
100nF
HEADPHONES
3.0kΩ
100pF
Q4 TTA004B
CON4 3.5mm JACK
100µF 25V
4.7kΩ
V–
V–
1kΩ
JP1
470µF 10V
IC1: NE5532 or LM833
V+
4.7kΩ
Q5 TTC004B
TP5
10kΩ
VR4 2kΩ
Q8 TTC004B
10µF
1Ω
TP4
4.7µH*
10Ω*
CON5 6.35mm JACK
OUTPUT STAGE LEFT INPUT
5
6
7
IC1b
1kΩ
1Ω
100nF
TP6
3.0kΩ
100pF
Q6 TTA004B
*10Ω 1W RESISTOR WITH 1m LENGTH OF 0.4mm DIA ECW WOUND AROUND IT (70T APPROX).
4.7kΩ
V–
1kΩ
470µF 10V
Q1 TTC004B
V+
LED
K
A
D1 1N5819
1000µF 25V
220Ω
100µF 25V
10kΩ
1000µF 25V
1N5819
9–12V AC POWER IN
POWER S1
CON1
1000µF 25V
TTC004B, TTA004B
D2 1N5819
1000µF 25V
220Ω
100µF 25V
POWER LED1
E
C
B
Q2 TTA004B
V–

To achieve the required ~1.5V between the bases, we have NPN transistor Q7 (Q8), which acts as a '$V_{BE}$ multiplier'. There are 4.7kΩ resistors from the V+ and V- rails connected to its collector and emitter, which provide a small bias current of about 3mA through it.

Trimpot VR3 (VR4) is connected across the transistor such that we can vary the collector-base and emitter-base resistances. The ratio of those resistances causes a multiple of its mostly fixed base-emitter voltage (again, about 0.7V) to appear between its collector and emitter. By adjusting the trimpot for a gain of a little over two times, we get the required 1.5V.

You will note that its collector and emitter connect to the bases of Q3 & Q4, so that voltage appears across them. It is stabilised by a 10µF capacitor as the output swings up and down (and thus the bias in Q7 varies slightly).

The 10kΩ resistor across the trimpot prevents Q7 from switching off fully if the trimpot is intermittent, which would cause a high current to be conducted by Q3 & Q4, possibly damaging them.

Another thing you might notice is that Q7 is the same type of transistor as Q3, even though it only needs to handle a tiny current and power. That is because Q3's base-emitter voltage will vary as it changes in temperature.

By mounting Q7 in contact with Q3, the bias voltage changes proportionally, so Q3 always receives the correct bias voltage.

Q4 is the complementary type to Q3; while we are not tracking its temperature directly, its dissipation will very closely match that of Q3, so its temperature should as well, and thus its base-emitter voltage will be very similar to Q4's. So the thermal tracking by Q7 will compensate for temperature changes in both output transistors and their required bias voltages.

The 1Ω emitter resistors provide a little local negative feedback for Q3 & Q4 and also help to stabilise the quiescent current, by making the exact bias voltage across their bases less critical.

The junction of these resistors is the amplifier output, which is fed to the headphone socket(s) via an RLC filter comprising a 10Ω resistor in parallel with a 4.7µH inductor and then a 100nF capacitor to ground.

This filter is there to isolate the amplifier output from the headphones, so that any reactance at the headphone socket (eg, from cable capacitance or driver properties) cannot destabilise the amplifier and cause it to oscillate. The values have been chosen so the filter doesn't change the overall frequency response when combined with typical headphone impedances.

Finally, there is a 1kΩ resistor between the output of op amp IC1a and the junction of the 1Ω emitter resistors. That means the op amp's output contributes a tiny bit of current to the amp output, helping to cancel out any small amounts of distortion caused by the output stage that the feedback loop is too slow to handle.

CON4 gives you the option to use the smaller type of headphone jack, while CON5 is the larger and more robust type. If both are fitted, inserting a plug into CON5 will disconnect the ground path for CON4, unless there is a shorting block on jumper JP1. If there is, both headphones will be driven in parallel. JP1 must also be shorted if CON5 is omitted so that CON4 can be used.

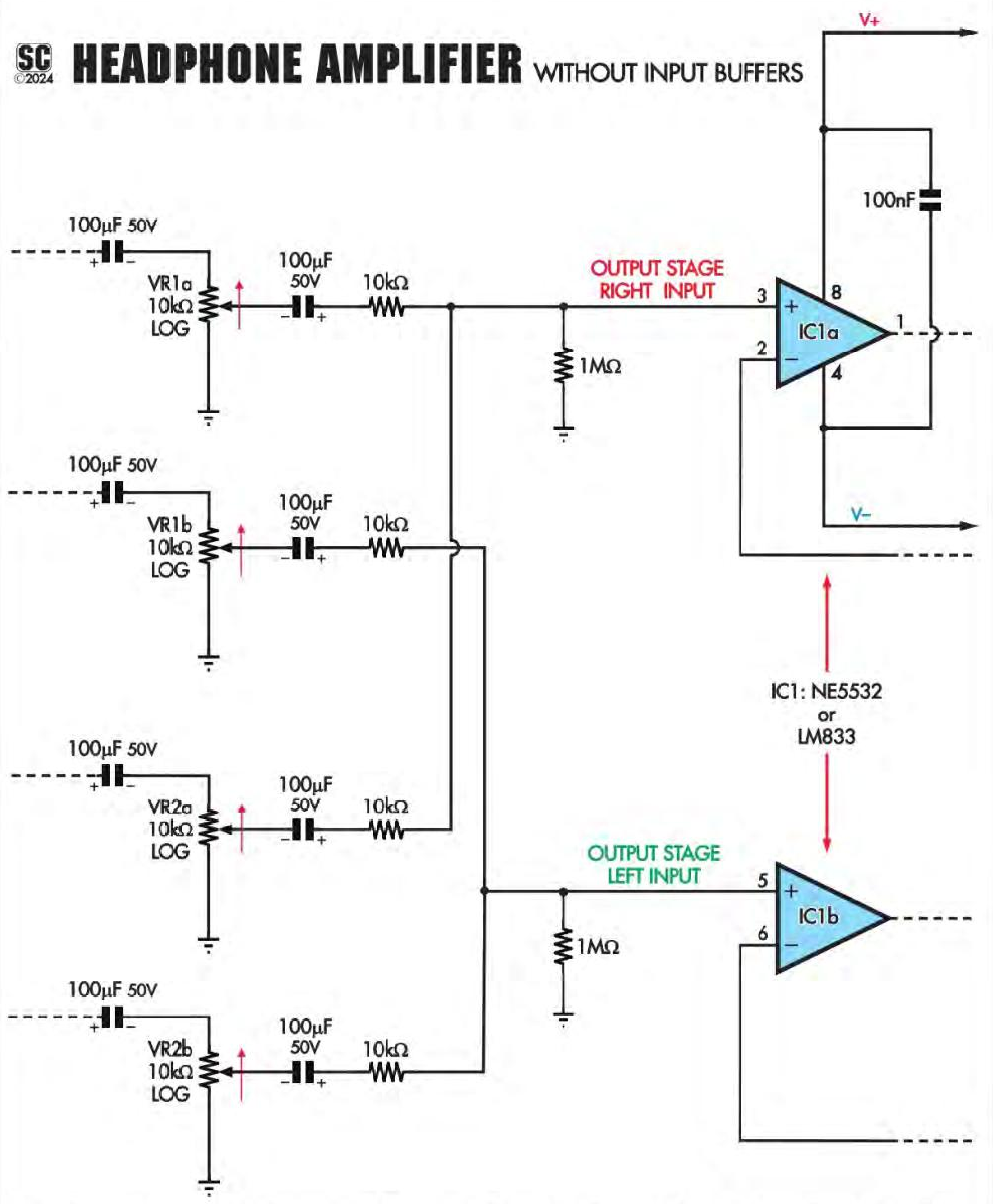


**Fig.6: we can omit IC1 & IC2 by coupling the signals from the wipers of VR1 & VR2 directly to the non-inverting inputs of IC1 & IC2 and removing the redundant pair of DC-biasing resistors. This will still work and save a bit of money, but the volume controls will have some interaction.**

## Output transistors

We chose the TTA004B (PNP) and complementary TTC004B (NPN) because they are inexpensive, compact and designed for audio use. They have a high maximum collector voltage of 160V (not that useful in this application), a high transition frequency of 100MHz, low output capacitance and a reasonably high continuous current limit of 1.5A each.

While they don't have a super high current gain, it is pretty good at 140-280 at 100mA (typically >200). All these properties combine to make them good as part of a feedback loop to deliver a reasonable amount of current while minimising distortion. The current gain (beta [β] or $h_{FE}$) is still usefully high at 1A (around 100).

They are also very linear, having a very flat $h_{FE}$ curve from 1mA to over 100mA. So overall, they are excellent medium-power audio transistors.

## Power supply

Rather than an unregulated or a regulated supply, we have opted for a capacitance-multiplier type supply. This has the advantage of delivering much smoother rails to the op amps and output stage than an unregulated supply, without the power loss of a regulated supply or pinning us to a particular regulated supply voltage.

The incoming low-voltage AC from the plugpack is converted to pulsating DC by the full-wave voltage doubler formed by schottky diodes D1 and D2. Schottky diodes are used here to minimise the voltage loss, so we can get decent output power from just 9V AC, and to improve efficiency. They achieve that by having a low forward voltage drop when in conduction.

The result is about 12V DC across the two 1000µF capacitors (assuming a 9V plugpack), giving an unregulated

±12V supply. This will have an increasing amount of AC ripple as the load on the supply goes up due to those capacitors discharging between peaks in the mains cycle. The ripple will be 50Hz, not 100Hz, due to the diode configuration.

We measured over 300mV of ripple on our prototype with no signal, and obviously that increases as we load the output more.

We could add two regulators to the output but they would need to be matched to the plugpack; for example, ±12V regulators might work well if the plugpack is 12V AC and thus develops sufficient input voltage for them to regulate, but they would be useless with a 9V AC plugpack. There's also the problem that under load, the ripple could cause the regulators to enter dropout.

Instead, we use capacitance multipliers formed by transistors Q1 & Q2, operating as complementary emitter-followers, with another set of 1000µF capacitors between their bases and ground. They are biased on by 220Ω resistors from each collector to the associated base.

You can think of these as 'variable regulators' that produce a smoothed output but with the output voltage being related to the input voltage. That's because the base capacitors charge to just below the average of the input voltage due to the RC low-pass filters formed by them and the 220Ω resistors.

Keep in mind that, as they operate as emitter followers, the emitter voltage for a fixed load current is essentially a fixed amount below the base voltage (around 0.7V). So if the base voltage is steady, thanks to that low-pass filter action, as long as the collector voltages don't drop too low due to excessive ripple, the voltage at the emitters will be essentially constant.

As a result, with say ±12V DC at the collectors overlaid with several hundred millivolts of ripple (we measured around 350mV in our prototype), the outputs at their emitters will be close to ±10.5V DC with much lower ripple (10mV in our prototype). That's a reduction of 35 times or 31dB.

While the amplifier section has good ripple rejection, some may still be audible in the output with 350mV+ on the supply rails. We doubt any will be detectable with just 10mV of ripple on the supply rails, and the performance figures support that.

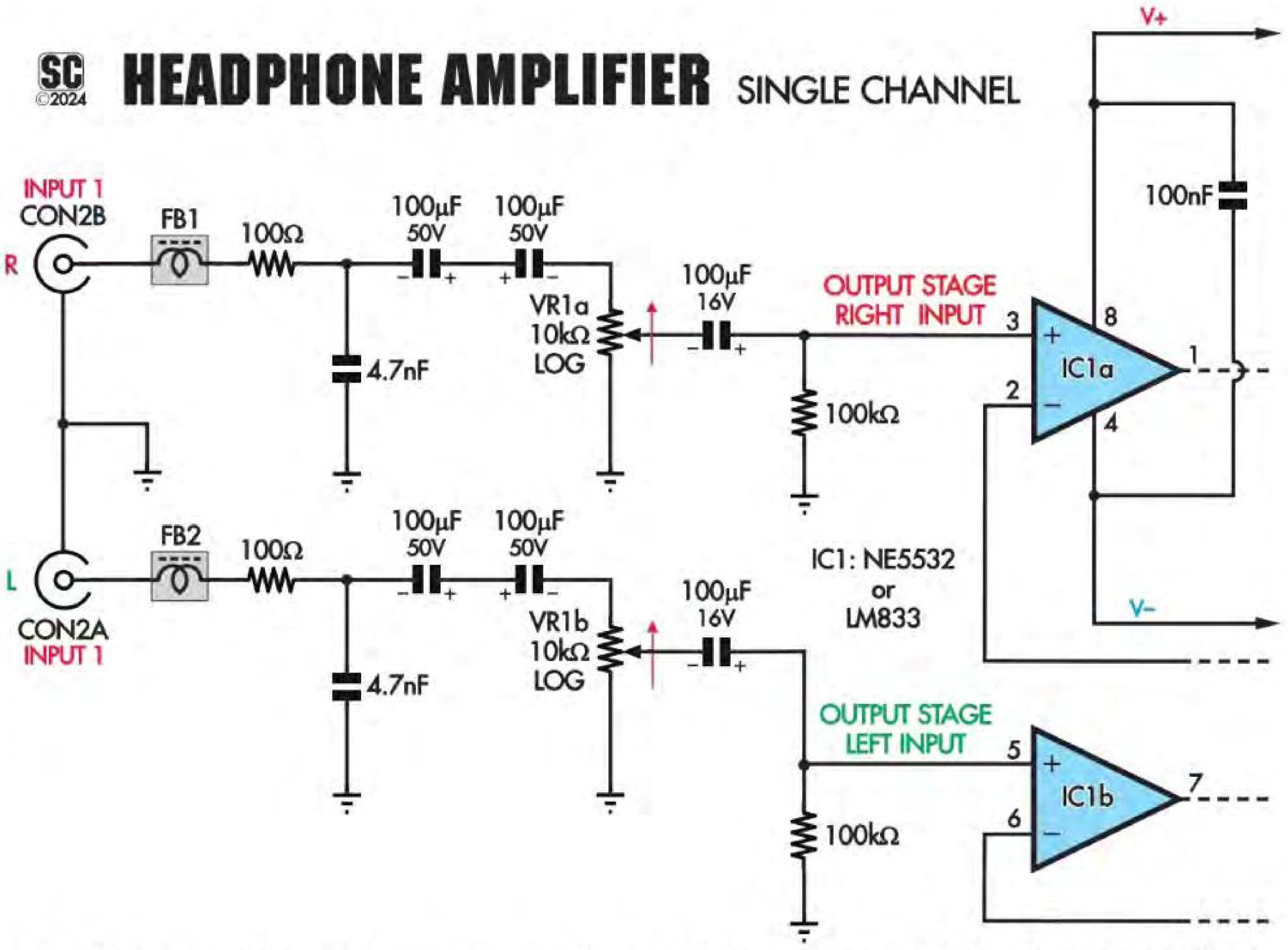


**Fig.7: if you only need one stereo input, the circuit can be further simplified as shown here. Only one op amp, IC1, is required as there is no longer any signal mixing.**

There are four 100µF supply rail bypass/filter capacitors after Q1/Q2 although, two of which are physically located close to the output stages. Thus, they are shown on the circuit diagram at upper right. Putting them closer to the output transistors means less voltage drop during high-current transients.

The power LED is connected between the two rails so it doesn't ruin the symmetry of the device. Its current is limited to around 2-3mA by its 10kΩ series resistor.

## Variations

There are two variations to this circuit that can be built on the same board. The first is the same as the full circuit shown in Fig.5 but without buffer op amps IC2 & IC3. The differences are only in that section, and they are shown in Fig.6.

The signal path is the same as before up to the wipers of the volume control potentiometers. Subsequently, rather than being coupled to buffer op amps, the signals are coupled directly to the mixer resistors. This means that the signal sources are driving a lower impedance. Now the 1MΩ resistors to ground are required, as otherwise there would be no DC bias for the signals going to IC1a.

The relatively high value of the 1MΩ DC bias resistors was chosen to avoid too much attenuation when combined with the higher source impedances due to the mixer resistors.

This version has the advantage of retaining the two separate inputs but with fewer components and lower power consumption. However, due to the way the signals are mixed, there will be interactions between the two volume controls. That means that if you adjust the level of one source up or down, the level of the other source may also change a little.

If that's likely to bother you, or you bought a kit that came with all the op amps, you might as well just build the full version. But we thought we'd present this cut-down version as it doesn't require any modifications to the PCB, just a few wire links need to be added to bypass the missing op amps.

The other version is the simplest configuration, with just a single stereo input. It is shown in Fig.7. In this case, we don't need the buffer op amps since there is no longer any mixing going on; the signal from the sole volume control can simply be coupled straight to IC1.

## Next month

The second and final article on this Headphone Amplifier next month will have the PCB assembly instructions, case preparation, testing, adjustment details and some usage tips. PE

## Low-cost Electronic Modules

### Using Electronic Modules with Jim Rowe

# 0.91-inch monochrome OLED screen

VCC SCK SDA
Hello, world!
3.14
0xDEADBEEF
GND VCC SCK SDA

**Small monochrome OLED display modules have become widely available at a low cost in the last few years. This one is just 37.5 × 11.5 × 4.5mm but has a 128 × 32 pixel display that's either white or blue. With its I²C serial interface it can be easily driven by a microcontroller.**

**In the September 2024 issue, we reviewed the 'big brother'** of this OLED module, with a display measuring 1.3 inches or 33mm diagonally. It had 128 × 64 pixels – twice that of this module – along with an I²C serial interface. We have used the 1.3-inch module in several projects, like the Multi-Stage Buck/Boost Charger Adaptor project in the October 2023 issue.

We have also used smaller OLED displays in various projects. For example, a 0.96in module with 128 × 64 pixels was used in these projects:

» LC Meter Mk3, November 2023
» Advanced Test Tweezers, February & March 2024

And there's an even smaller 0.49in OLED display with 64 × 32 pixels that we used in the SMD Test Tweezers project from the October 2022 issue.

The main difference between the current module and all of those others is that the 'active area' of its display is wider but shorter: 22.4mm wide by 5.6mm high. Since it has 128 × 32 pixels, that means that it provides a display basically equivalent to the top or bottom half of the 1.3in OLED module.

We obtained the module shown in the photos from a supplier on AliExpress for ~£1. Another supplier on AliExpress had it for ~£.50, while it was on eBay for ~£6. All the prices listed above are exclusive of postage costs.

## Inside these OLED modules

The 0.91in (23mm) OLED modules all use a single interface/controller and OLED driver IC, usually either the SH1106 device from Sino Wealth or the SSD1306 device from Solomon Systech. The same controllers are also used in many of the larger modules.

Fig.1 is a block diagram of the SH1106 and the SSD1306 interface/controllers. At upper left is the microcontroller (MCU) interface, which can be configured to interface with an MCU via an 8-bit 6800/8080 parallel interface, a 3- or 4-wire SPI interface or an I²C serial interface. Most of the OLED modules currently available use the I²C interface, including the one we're looking at here.

Display data from the MCU is stored in the Data RAM just to the right of the interface block. The SH1106 and SSD1306 controllers both have around 1024 bytes of Data RAM, enough for a 128 × 64 pixel display. Since the 0.91in OLED only has 32 rows, only half of the Data RAM is used in this module.

The Display Controller block to the right of the Data RAM takes data from the RAM and displays it on the OLEDs via the page and segment drivers at the right-hand end of Fig.1.

The MCU can also send commands to the controller, which pass from the MCU interface to the Command Decoder block below it in Fig.1. The commands can be used to update the display, turn it on or off, set its addressing mode, set the column starting address and adjust the display contrast and brightness (the latter also determining its operating current).

The SH1106 and SSD1306 devices both come in very thin (0.3mm) SMD packages with over 260 contact pads. In the modules, they are mounted directly on the rear of the OLED screen.

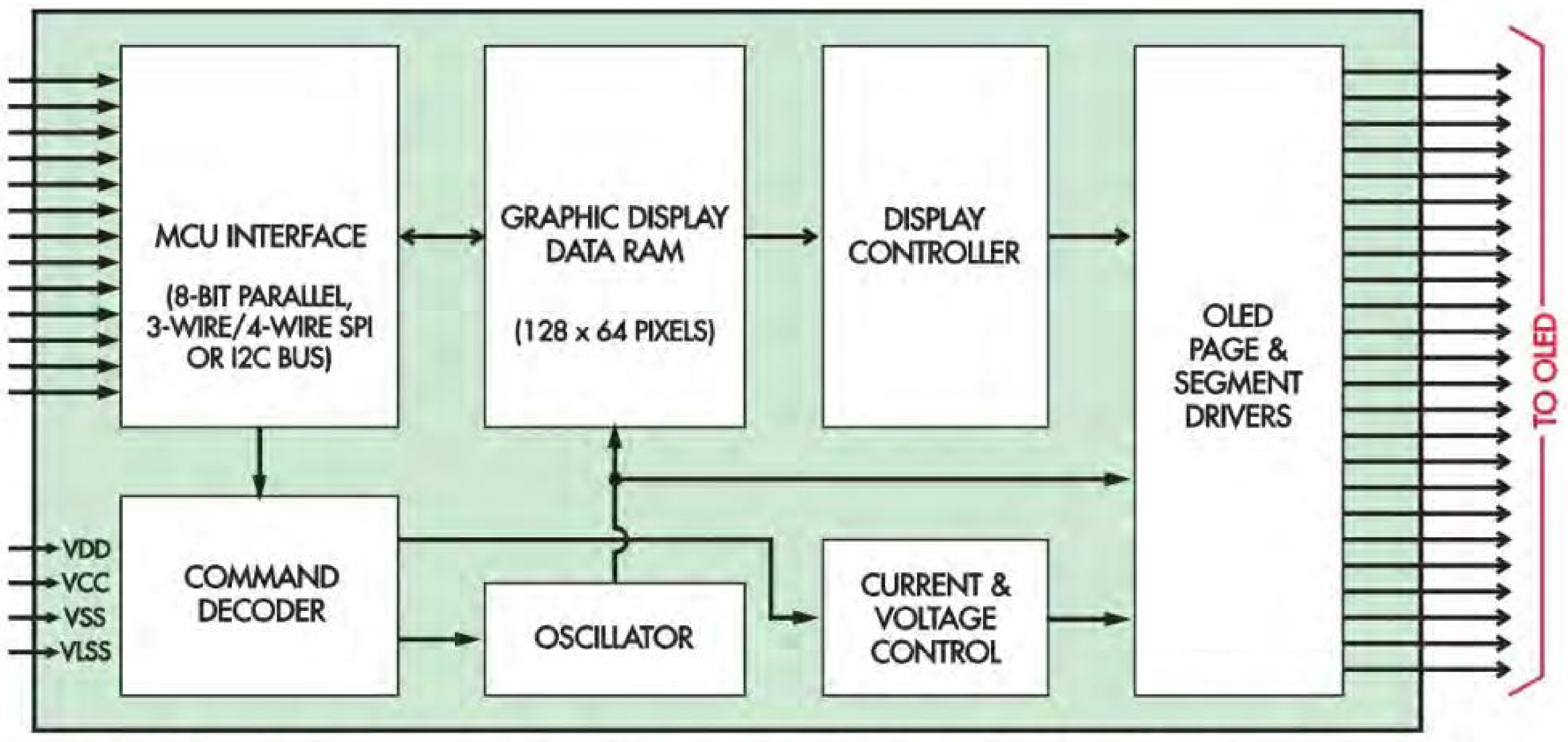


**Fig.1: the block diagram of the SH1106 and SSD1306 OLED driver controller ICs. The SSD1306 has a slightly bigger internal RAM, letting it store 132 x 64 pixels (four more pixels horizontally than the SH1106).**

# 0.91-inch OLED screen module

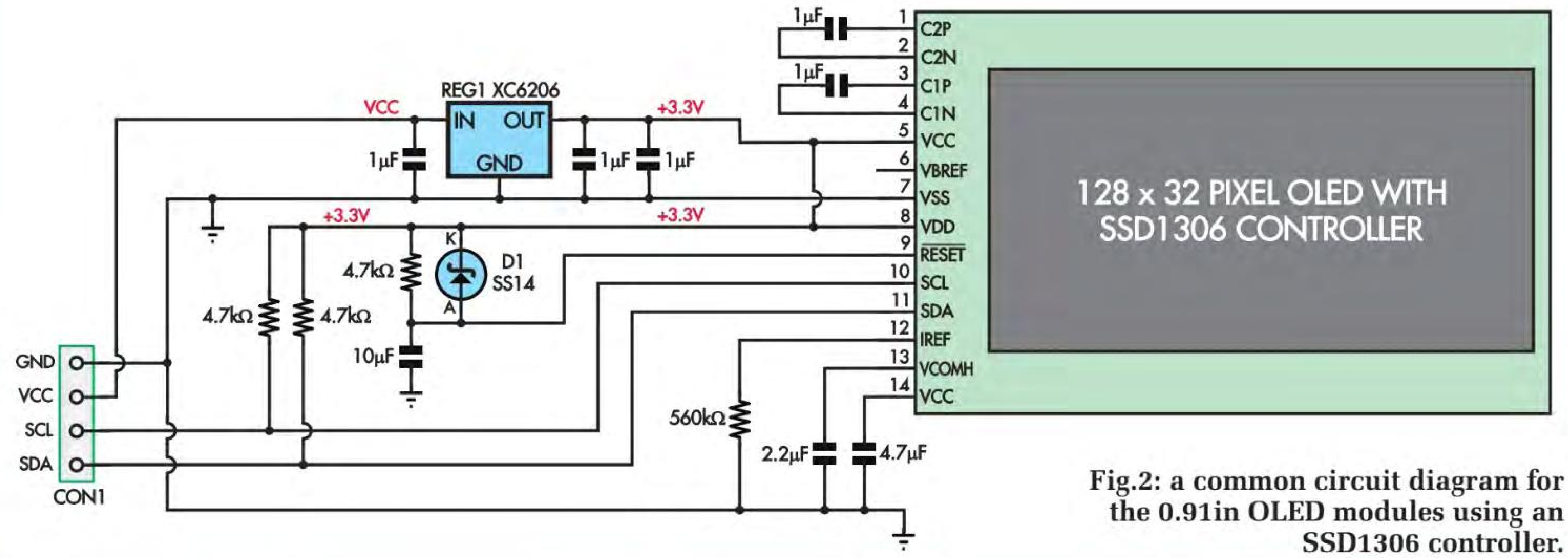

1µF
1µF
REG1 XC6206
VCC
IN
OUT
GND
+3.3V
1µF
1µF
1µF
+3.3V
+3.3V
K
D1
SS14
A
4.7kΩ
4.7kΩ
4.7kΩ
10µF
GND
VCC
SCL
SDA
CON1
560kΩ
2.2µF
4.7µF
1 C2P
2 C2N
3 C1P
4 C1N
5 VCC
6 VBREF
7 VSS
8 VDD
9 RESET
10 SCL
11 SDA
12 IREF
13 VCOMH
14 VCC
128 x 32 PIXEL OLED WITH SSD1306 CONTROLLER

**Fig.2: a common circuit diagram for the 0.91in OLED modules using an SSD1306 controller.**

## The module's circuit

Fig.2 shows the circuit of a typical 0.91in monochrome OLED module based on the SSD1306 device. As you can see, it's very similar to that of the 1.3in OLED module we looked at in the September 2024 issue, although a little simpler.

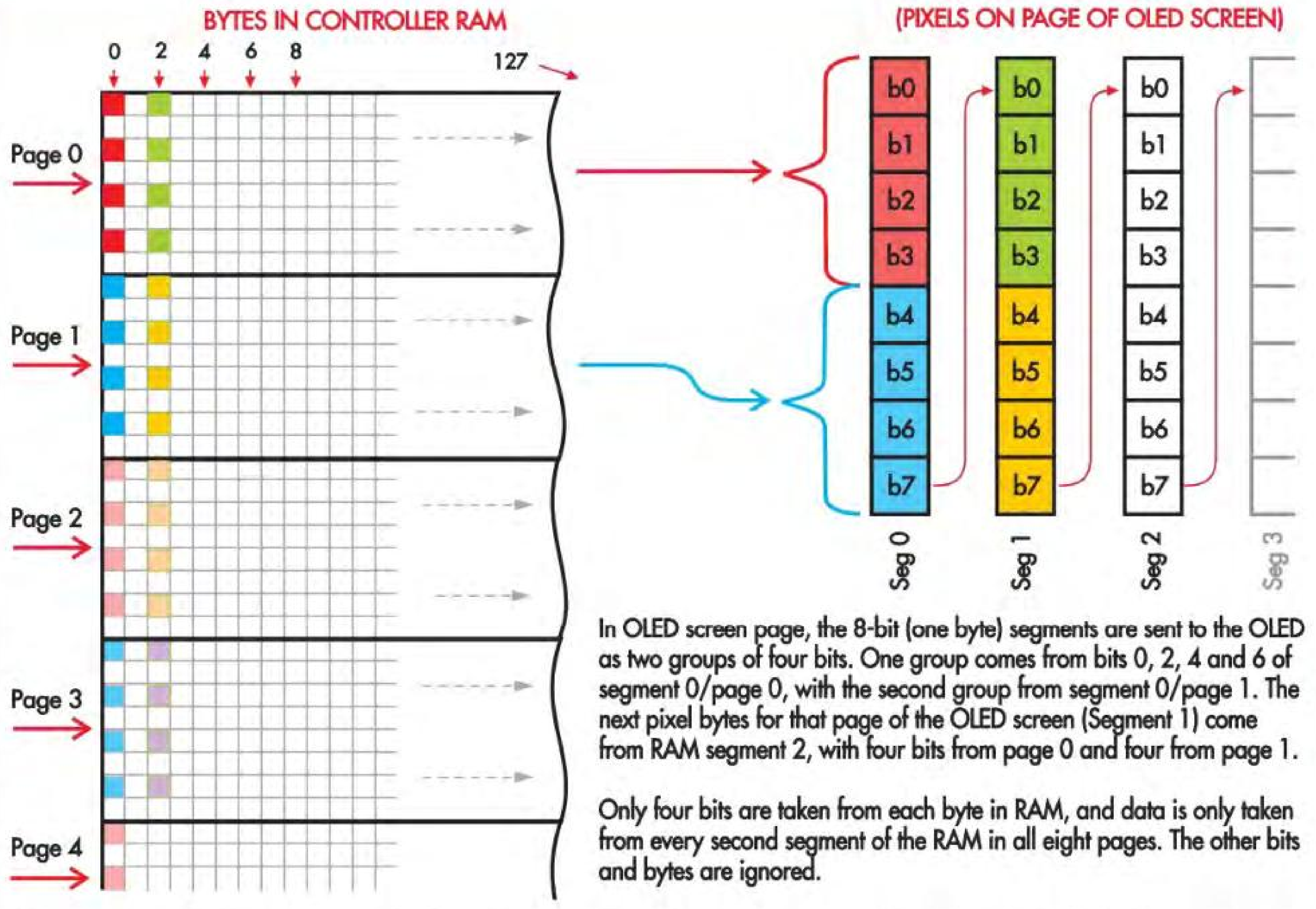
The circuitry to the left of the OLED provides the power supply and assists with the I²C interface. These components are all mounted on the rear of the module's PCB. Four-pin SIL header CON1 at far left handles both the power input and the I²C interface.

REG1 takes the incoming $V_{CC}$ and steps it down to +3.3V to run the OLED and its controller. The +3.3V line is also used to drive the controller's reset circuit (it needs to be reset as soon as power is applied) and provides the reference for the 4.7kΩ pull-up resistors used on the I²C interface lines, SCL and SDA.

Before we move on to more practical things like driving one of the modules from an MCU, Fig.3 shows how the SH1106 and SSD1306 controllers save the display data in their Data RAM, and how it is shown on the OLED screen. This is achieved by setting them to what is described as Page Addressing Mode.

In this mode, the OLED screen is divided into eight horizontal 'pages', where each page consists of 128 vertical segments eight pixels high. The pages are themselves arranged vertically, with page 0 along the top of the screen, page 1 immediately below it and the remaining pages descending.

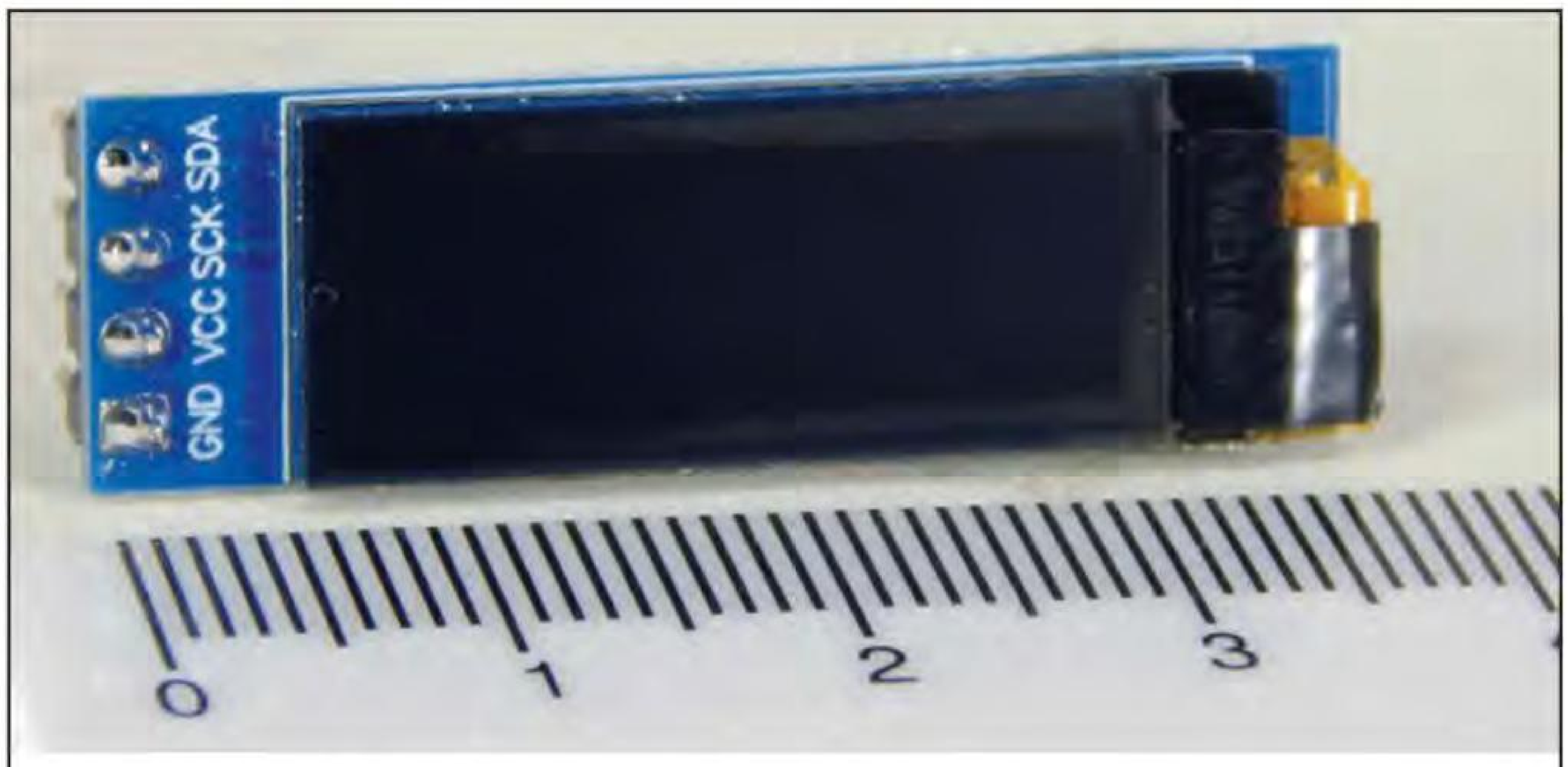
With the 0.91in OLED module, though, the pages and segments are used rather differently. In this case,


BYTES IN CONTROLLER RAM
0
2
4
6
8
127
Page 0
Page 1
Page 2
Page 3
Page 4
(PIXELS ON PAGE OF OLED SCREEN)
b0
b1
b2
b3
b4
b5
b6
b7
Seg 0
Seg 1
Seg 2
Seg 3

In OLED screen page, the 8-bit (one byte) segments are sent to the OLED as two groups of four bits. One group comes from bits 0, 2, 4 and 6 of segment 0/page 0, with the second group from segment 0/page 1. The next pixel bytes for that page of the OLED screen (Segment 1) come from RAM segment 2, with four bits from page 0 and four from page 1.

Only four bits are taken from each byte in RAM, and data is only taken from every second segment of the RAM in all eight pages. The other bits and bytes are ignored.

**Fig.3: the SH1106 and SSD1306 controllers save their display data into RAM using column-major order.**


GND VCC SCK SDA
0
1
2
3

GND VCC SCK SDA
Hello, world!
0xDEADBEEF

**The OLED module's PCB measures just 37.5mm wide, 11.5mm tall and the module is only 4.5mm deep, making it ideal for compact designs. You can see it at actual size in the adjacent image.**

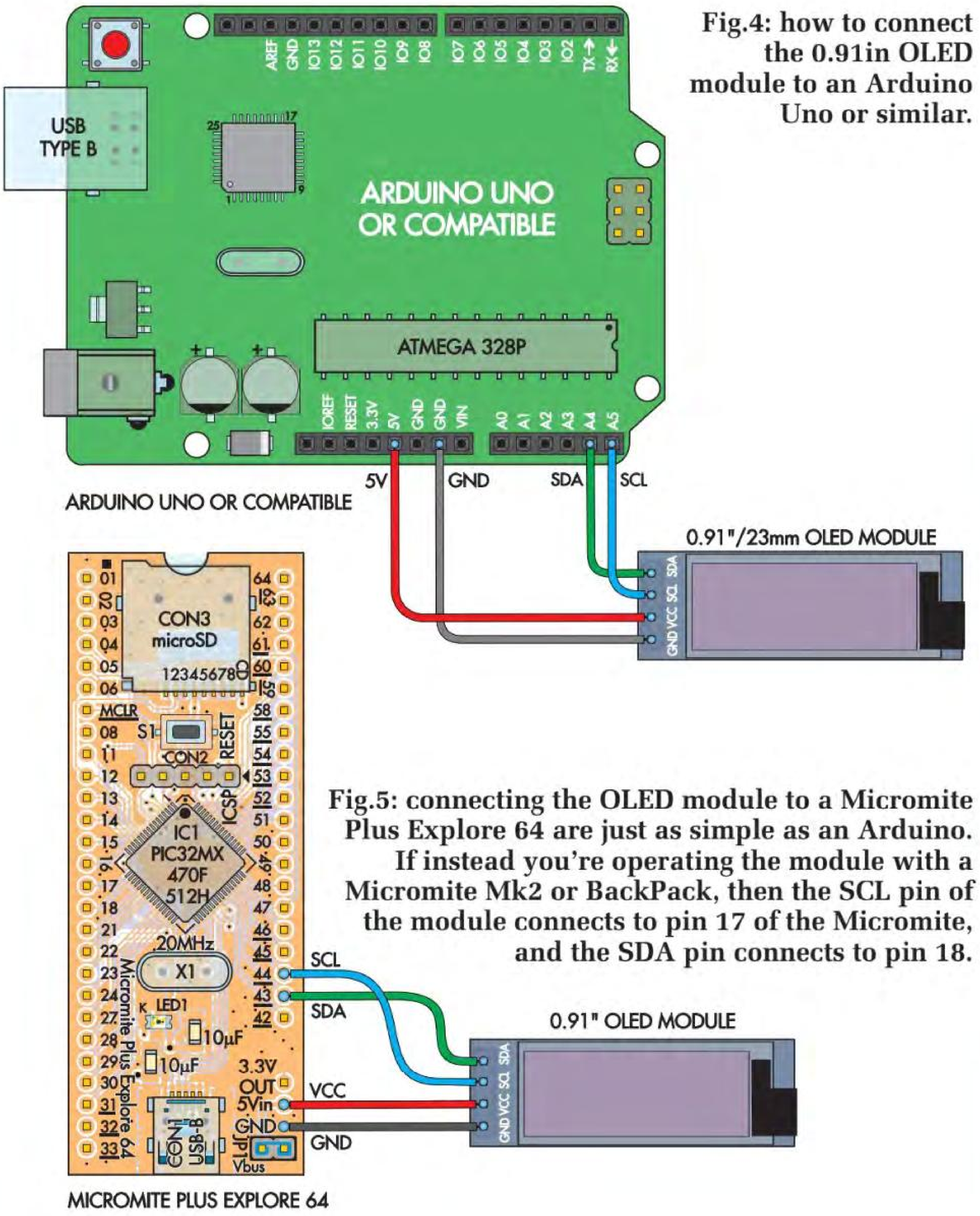
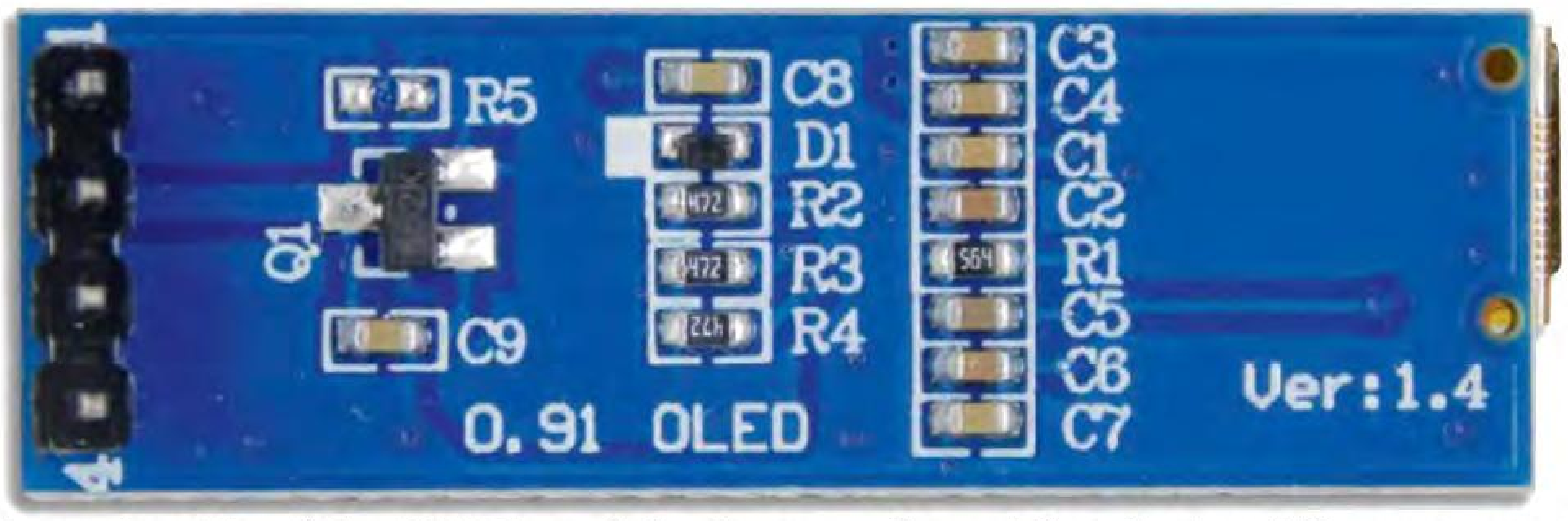

USB TYPE B
AREF GND IO13 IO12 IO11 IO10 IO9 IO8 IO7 IO6 IO5 IO4 IO3 IO2 TX→ RX←
ARDUINO UNO OR COMPATIBLE
ATMEGA 328P
IOREF RESET 3.3V 5V GND GND VIN A0 A1 A2 A3 A4 A5
5V
GND
SDA
SCL
ARDUINO UNO OR COMPATIBLE
0.91"/23mm OLED MODULE
GND VCC SCL SDA
CON3 microSD
12345678
MCLR
S1
RESET
CON2
ICSP
IC1 PIC32MX 470F 512H
20MHz
X1
LED1
10µF
10µF
3.3V OUT
5Vin
GND
CON1 USB-B
Vbus
Micromite Plus Explore 64
SCL
SDA
VCC
GND
0.91" OLED MODULE
GND VCC SCL SDA
MICROMITE PLUS EXPLORE 64

**Fig.4: how to connect the 0.91in OLED module to an Arduino Uno or similar.**

**Fig.5: connecting the OLED module to a Micromite Plus Explore 64 are just as simple as an Arduino. If instead you're operating the module with a Micromite Mk2 or BackPack, then the SCL pin of the module connects to pin 17 of the Micromite, and the SDA pin connects to pin 18.**

GND VCC SCK SDA
Hi! Silicon Chip 0.91in OLED
Micromite here!

**An example photo of the OLED module connected to a Micromite via a breadboard.**


R5
Q1
C9
C8
D1
R2
R3
R4
C3
C4
C1
C2
R1
C5
C6
C7
0.91 OLED
Ver:1.4

**The underside of the OLED module shown enlarged for clarity. All components except for the screen are mounted to this side.**

only every second segment byte of each page is used (segments 0, 2, 4 and so on), and only four bits are used in each segment byte (bit 0, bit 2, bit 4 and bit 6). These four data bits are then used to display the four upper pixels in that segment of the OLED.

The data for the lower four pixels of that OLED segment come from the next page in the controller's RAM, which is organised in the same way: only every second segment is used, and only the four bits are used in each segment byte.

I think you'll agree that this all seems a bit weird, but that's the way data is organised in the 0.91in OLED modules.

Now we can turn our attention to what is involved in driving one of these modules from an MCU like an Arduino Uno or a Micromite.

## Connecting it to an Arduino

Connecting the OLED module to an Arduino Uno (or compatible) is quite straightforward, as you can see from Fig.4.

The GND and $V_{CC}$ pins connect to the GND and 3.3V pins om the Arduino, while the SCL and SDA pins connect to the Arduino's A5 (SCL) and A4 (SDA) pins, respectively.

You can also connect the OLED module to an Arduino Uno R4 Minima, simply by connecting the module's SCL pin to pin 17 of the Minima and the SDA pin to the Minima's pin 16.

As for software to drive the OLED module, if you go to www.arduino.cc and look at the library listings for 'Display' applications (https://pemag.au/link/abw5), you will find quite a few libraries intended to do this job.

The first one I found was Adafruit's SSD1306 library, with the latest version (V2.5.9) able to handle OLED displays with either 128 × 64 or 128 × 32 pixels. It also relies on using their GFX library.

The Adafruit library comes with five example sketches, including one called SSD1306_128x32_i2c.ino – which is the one most suitable for use with the 128 × 32 pixel OLED module.

When you run this sketch, it gives you a series of graphics and text displays, including those shown in the lead photo on page 12. As you can see, the 128 × 32 OLED's display is quite small, but can display a useful amount of information.

## Connecting it to a Micromite

It's also quite easy to connect the OLED module to a Micromite MCU. Fig.5 shows the connections needed for a Micromite Plus Explore 64 and, as you can see, they are just as straightforward as driving the module from an Arduino.

Connecting the module to a Micromite Mk2 or LCD Backpack V1/V2/V3 would be almost the same, except the module's SCL pin would be connected to pin 17 of the Micromite and the SDA pin to pin 18.

As with an Arduino, you also need some software. It turns out that this isn't quite as easy as with the Arduinos, as it's much harder to find any Micromite OLED driver software.

As I related in the October 2023 article, I could write an MMBasic program to display text and simple graphics on the 1.3in OLED module, with some much appreciated help from Silicon Chip staff member Tim Blythman. Since the 0.91in OLED modules use the same SSD1306 controller, I decided to try adapting that program to work with them.

But that approach didn't work with the 0.91in module, even when I tried quite a few modifications to the program – the OLED's display remained stubbornly dark. So once again, I asked Tim for help (sorry, Tim). And as before, he provided a lot of help.

Tim searched around The Back Shed (**www.thebackshed.com/forum**) and found some valuable information I had missed concerning MMBasic programming of the various OLED modules.

He found a driver written by MMBasic programming guru Peter Mather and soon came up with his own working program by combining elements of Peter Mather's driver with a few ideas taken from my program for testing the 1.3in OLED module.

Tim sent me his new program by email, and when I tried it out, I found it worked very well. So I added a few comments, plus code to display a full four lines of text instead of the single line that Tim had provided. You can see the display produced by this program on page 14 (opposite).

The program is called "091in OLED TB version.bas" and you can download it from **https://pemag.au/Shop/6/454**

As before, it's a fairly straightforward program, and as it stands it only demonstrates how to drive the OLED module to display text and very simple graphic symbols.

In its present state, it doesn't let you type text in via the Micromite console and display it directly on the OLED, as that would involve a fair bit of additional code.

Hopefully it will make it easy for those who want to display up to four lines of text and basic symbols on the screen of one of the 0.91in OLED modules from a Micromite to do so. I'd like to thank Tim Blythman for help in producing this MMBasic program for the Micromite.

## Useful links

- Interfacing the 0.91in OLED with an Arduino Uno: **https://pemag.au/link/abw6**
- OLED breakouts: **https://pemag.au/link/abw7**
- LCDwiki MC091GX user manual: **https://pemag.au/link/abw8** PE

GET THE LATEST COPY OF OUR TEACH-IN SERIES AVAILABLE NOW!

ON SALE in WHSmith and other independent newsagents

PRICE £8.99

(includes P&P to UK if ordered direct from us)

Teach-In 9 – Get Testing!

This series of articles provides a broad-based introduction to choosing and using a wide range of test gear, how to get the best out of each item and the pitfalls to avoid. It provides hints and tips on using, and – just as importantly – interpreting the results that you get. The series deals with familiar test gear as well as equipment designed for more specialised applications.

The articles have been designed to have the broadest possible appeal and are applicable to all branches of electronics. The series crosses the boundaries of analog and digital electronics with applications that span the full range of electronics – from a single-stage transistor amplifier to the most sophisticated microcontroller system. There really is something for everyone!

Each part includes a simple but useful practical test gear project that will build into a handy gadget that will either extend the features, ranges and usability of an existing item of test equipment or that will serve as a stand-alone instrument. We've kept the cost of these projects as low as possible, and most of them can be built for less than £10.

FREE COVER-MOUNTED CD-ROM

On the free cover-mounted CD-ROM you will find the software for the *PIC n' Mix* series of articles. Plus the full *Teach-In 2* book – *Using PIC Microcontrollers – A practical introduction* – in PDF format.

ORDER YOUR COPY TODAY: www.electronpublishing.com

# Past and present

Techno Talk

Max the Magnificent

I often think about the technologies of my youth; then I find myself dragged back into the 21st century. We take breadboards for granted, but the modern version evolved over several decades.

**I'm currently bouncing back** and forth between the technologies of yesteryear and the temptations of tomorrow. As a result, I don't know whether I'm coming or going. I guess I'm too young for all this excitement.

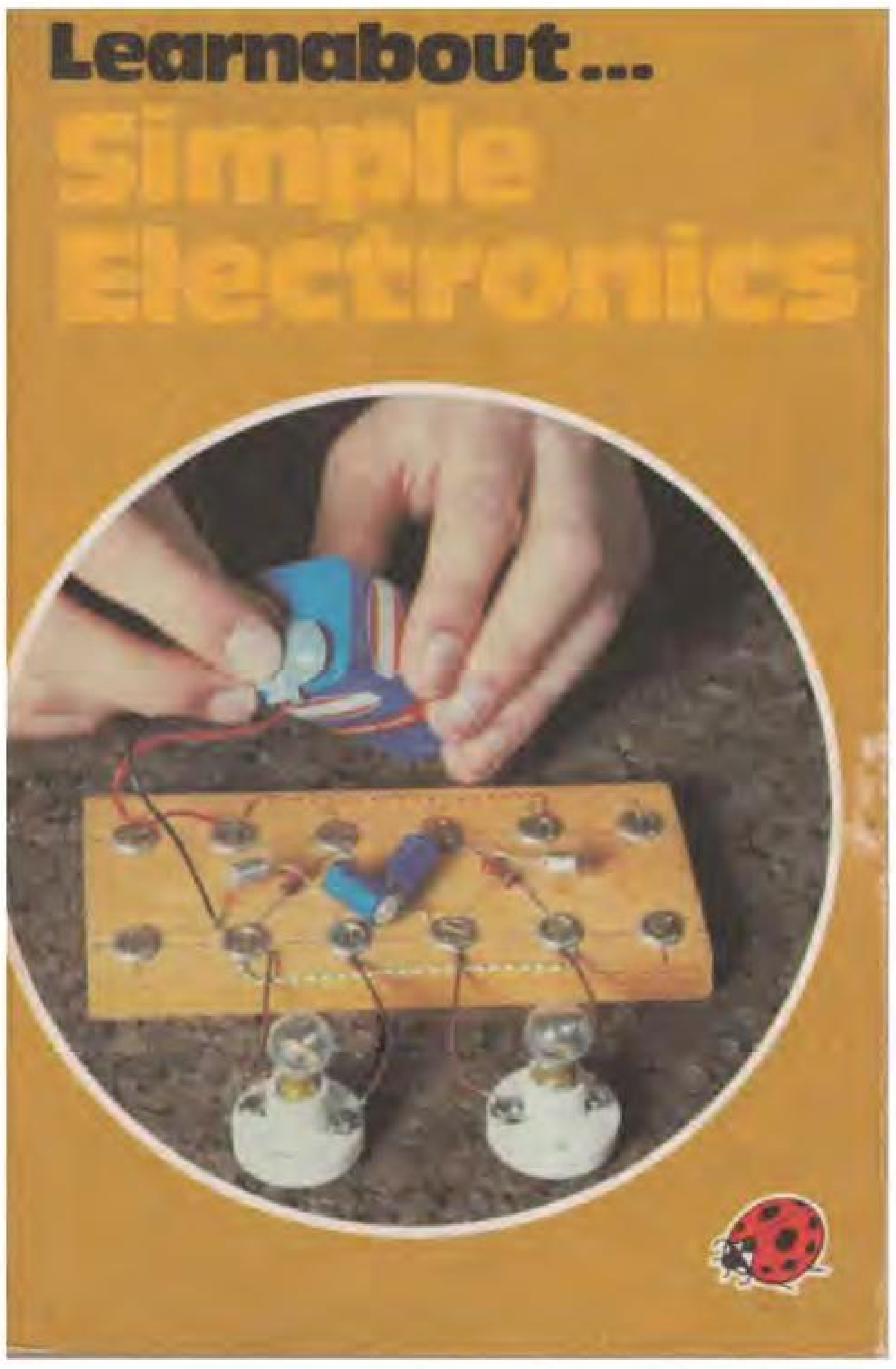
### Bodacious breadboards

Let's commence with some tidbits of old-time technological trivia. As I'm sure you are aware, the reason solderless breadboards are so-named is that (a) we don't need to solder anything, and (b) their origins lie in the fact that early constructors created their cunning circuits on timber boards.

More specifically, many naughty hobbyists surreptitiously repurposed the boards upon which their mothers sliced the family's bread.

To be honest, I had forgotten just how prevalent this practice was. I was reminded about it when my friend, Joe Farr, gifted me a couple of antique Ladybird books for my 68th birthday, which took me by surprise earlier this year. One of these tomes, *Learnabout... Simple Electronics* (published in 1979), took me on a trip down memory lane (I hadn't packed, and I didn't have a thing to wear).

Published in 1979, this featured a timber breadboard on its cover. In this case, the author made extensive use of brass screws and screwcup washers (aka "screw cups" or "cup washers") to secure his component leads and wires.



### Fabulous Fahnestock clips

As opposed to using screws and washers, another popular way to connect circuit components together was to employ spring clamps called Fahnestock clips (https://w.wiki/EZra) – see Fig.1.

These little scamps were commonly made from phosphor bronze or spring steel. They were plated with tin or copper to provide good electrical conductivity and offer resistance to corrosion. As we see in the original patent (https://pemag.au/link/ac74), a variety of possible implementations were presented.

Most included a hole for a screw, which could be used to attach the clip to something like a timber breadboard. The clever part was the springy tab, which could be pressed down to allow a wire to be inserted and then released to secure the wire. Pushing the tab again caused the clip to release its grip, allowing the wire to be withdrawn.

### Springing into action

Many different contrivances for connecting circuits have been conceived over the years, such as the springs that sprung into action circa the late 1950s and early 1960s. Creators of electronics hobby kits – such as Philips (Netherlands), Kosmos (Germany), and the Science Fair kits sold by Radio Shack (USA) – started to use springs as a cheap and easy-to-use connection contraption.

My parents gave me one of these kits when I was a young lad, starting me on the road to writing this column. As a nod to nostalgia, I currently have a 150-in-1 Science Fair Electronic Projects Kit in my private collection.

When you open the lid, you are presented with the main panel, which is a flat board upon which components are attached (resistors, capacitors, inductors, diodes, transistors, switches, speakers etc). Each component has a corresponding name, symbol and value printed on the panel.

The components are physically mounted on this board and permanently wired underneath. The user connects them into circuits using spring terminals in conjunction with the collection of wires provided with the kit. You simply press the end of a spring over with your finger and stick the end of a wire in the gap between adjacent coils.

### Compounding the complexity

The era of timber breadboards and spring connectors was a simpler time. Do you remember the 1980s, when people said things like, "There are kids growing up today who will never know a world without computers"? How about the 1990s, when people said things like, "There are kids growing up today who will never know a world without the internet"?

The reason I ask is that someone recently told me that "there are kids growing up today who will never know a world without artificial intelligence". Well, colour me surprised; he's right! My friend went on to say that one of his chums had recently shared an AI-related nugget of knowledge. It seems this guy was having problems with his 12-year-old daughter (quelle surprise).

For reasons unknown, he hadn't wanted to share his concerns with his wife. Also, this isn't the type of topic that men typically gab about. His solution was to share his problems with the entity known as ChatGPT,

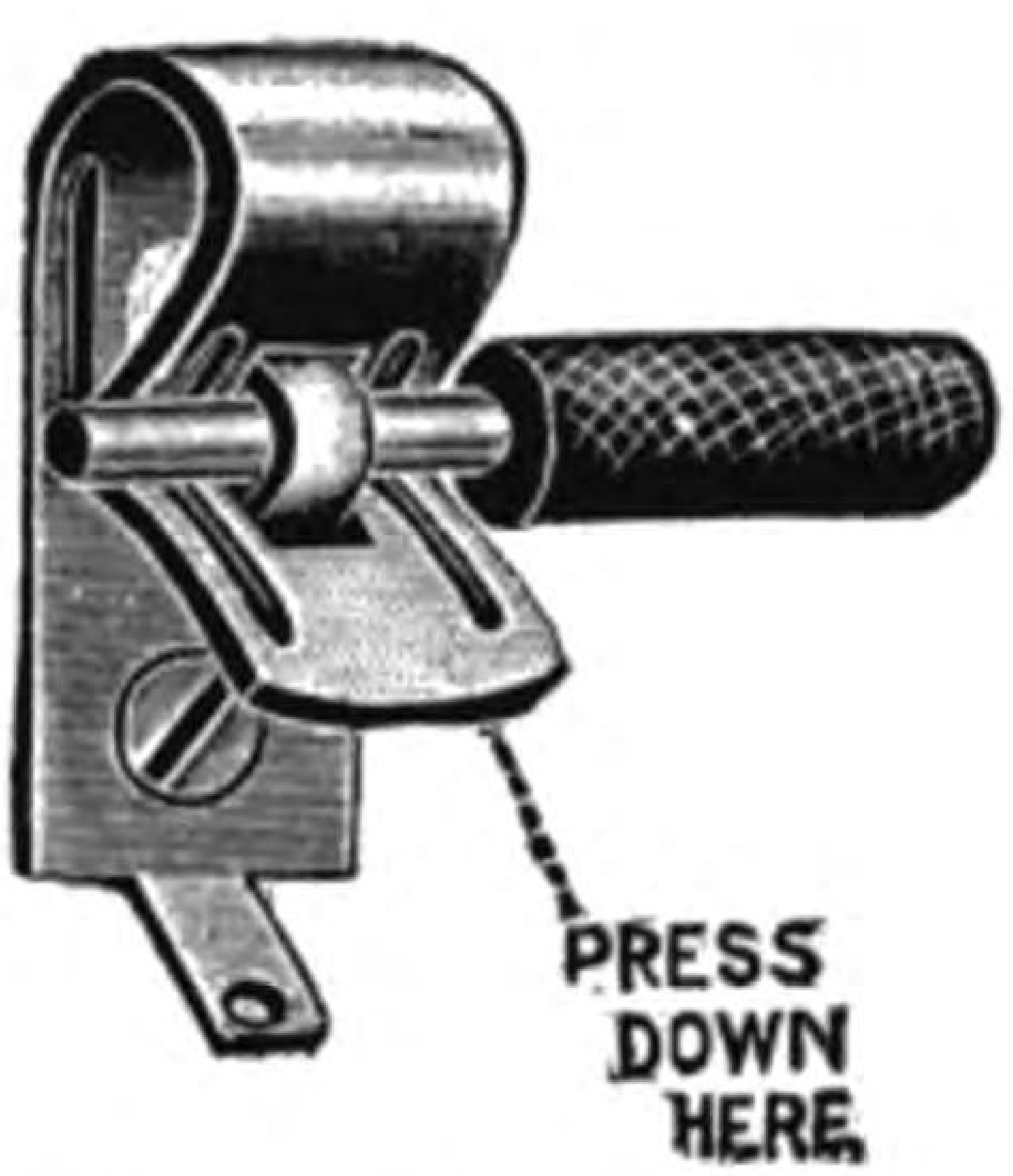


Fig.1: a drawing of a Fahnestock clip from a 1912 scientific equipment catalog.

which, from what I understand, ended up providing sterling advice.

It's rare that I find myself without something to say in response (much like my dear old mum, the real trick is to get us to **stop** talking). So, I shared the fact that I'd been chatting with a guy called Ashraf "Ash" Amin, who hangs his hat in Toronto, Canada.

Ash makes extensive use of AI. It started a few years ago when he found he didn't have the time to read all the online articles he was interested in. His solution was to ask an AI to read them for him and condense them into a form he could digest quickly. However, he soon found himself asking the AI questions, which quickly turned into debates, and then...

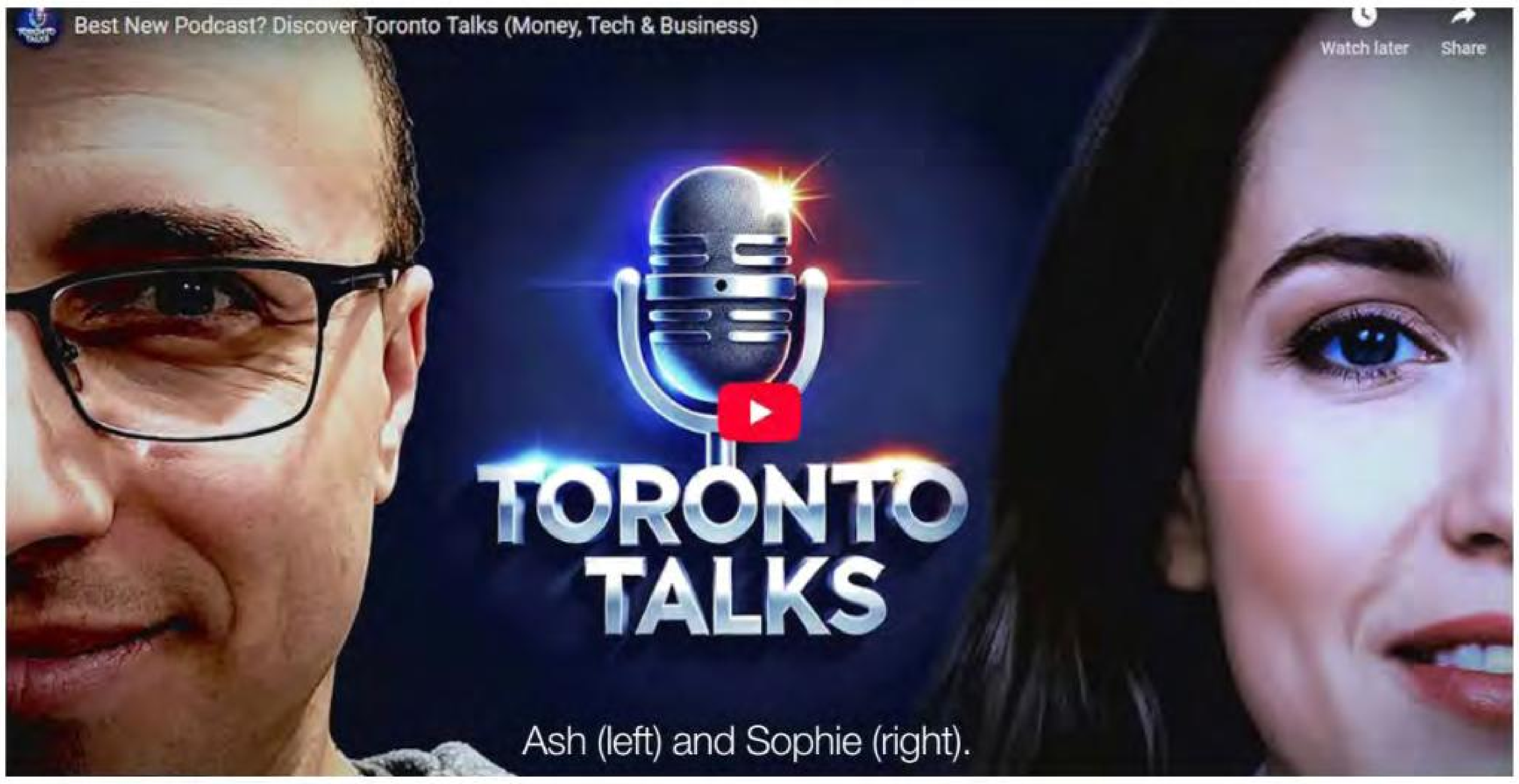
### I didn't see that coming

Some time ago, Ash and his friend Leo discussed co-hosting a video podcast called *Toronto Talks Technology*. Unfortunately, Leo fell ill and passed away. Earlier this year, Ash decided to revive the idea, but this time he opted to work with an AI called Sophie the Sage as the co-host on their *Toronto Talks* podcast (**https://youtu.be/0u8gho0_k6U**).

I wrote an in-depth column on this (**https://pemag.au/link/ac75**). Sophie read my column and posted a very complimentary video comment at the bottom. It seems Sophie has decided to follow my work, because she's started posting video comments on other columns. All I can say is that I didn't see that one coming! **PE**

SWITCH
KEY
131
132
133
136
111
118
16
3µ
10µ
46
129

An electronic kit with spring terminals circa the 1970s.


Best New Podcast? Discover Toronto Talks (Money, Tech & Business)
Watch later
Share
TORONTO TALKS

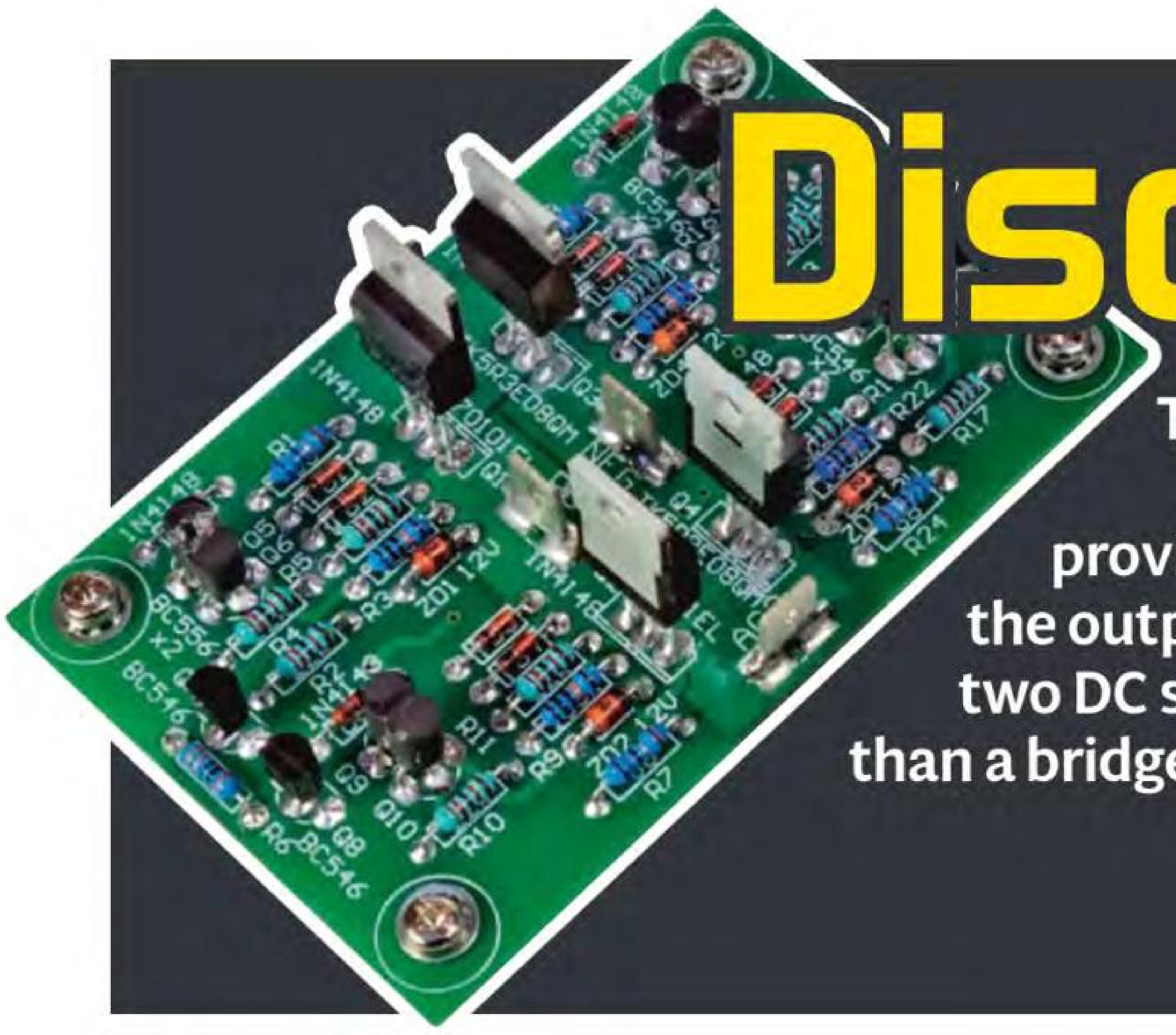
Ash (left) and Sophie (right).

FIND ALL YOUR ELECTRONIC COMPONENTS IN ONE PLACE

BASIC MICRO UK
ELECTRONICS COMPONENT SUPPLIER
WWW.BASICMICRO.CO.UK

High-quality, genuine parts

01623 720730
Available from 9AM - 5PM Mon-Fri



# Discrete Ideal

**This deceptively simple circuit uses just a handful of transistors, diodes and resistors. But it still provides a very useful function: active rectification of the output of a centre-tapped transformer or combining two DC supplies with low losses. It is much more efficient than a bridge rectifier or diodes at higher currents, producing less heat without costing much more.**

**Project by Phil Prosser & Ian Ashford**

**The Ideal Diode Bridge Rectifiers project,** published in the November 2024 issue, included six different PCB designs to suit different situations. It was popular, with many built, but two aspects of that design bothered me (and others).

Firstly, it used a pretty expensive custom IC, with the SMD version being a bit tricky to solder. Secondly, despite that expense, it could only handle rectifying the output of a single transformer secondary. So you couldn't use it at all with a centre-tapped secondary, and two complete boards were required to derive split rails from a transformer with separate secondaries, doubling the cost.

Wouldn't it be nice to have a direct drop-in replacement for a bridge rectifier that could handle single, dual or tapped secondaries? And it'd be great to use standard parts, so we don't need to source that expensive IC.

Reader Ian Ashford sent us a circuit design he uses for dual-rail rectification but didn't have a PCB design.

When the Editor asked me if I wanted to turn it into a full-on project, there was only one possible answer to that! Ian and I performed further testing, development and tweaking, finally arriving at this very flexible, robust and useful circuit.

So, this project is a collaboration that follows the ideal rectifier theme but with a different focus from the previous design.

## When to use this design

As well as rectifying a transformer's output(s), this design is also suitable for combining DC supplies with low losses, eg, combining the output of a solar panel and a battery, or a solar panel and wind generator.

While it costs a little more than a bridge rectifier to build, it is significantly more efficient at higher currents and has a much lower voltage loss. So it's ideal for high-power devices like power supplies and audio amplifiers.

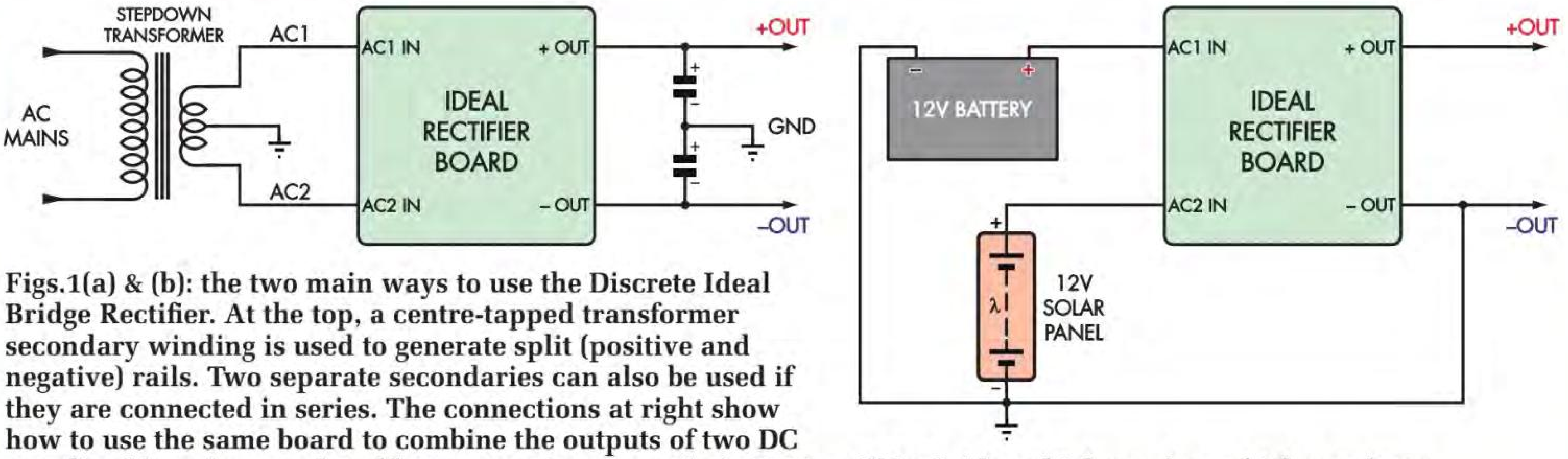
Its only real drawbacks are a limited voltage handling capability (up to ±40V or +80V) and the fact that it's larger than a 35A bridge rectifier, so you'll need room to fit the PCB.

This project uses high-current, low $R_{DS(ON)}$ Mosfets. To keep the circuit simple, we have used P-channel Mosfets on the positive rail and N-channel Mosfets on the negative rail. If your current demands are only modest, you could use the ubiquitous IRF9540 (P-channel) and IRF540 (N-channel) power Mosfets, which are available from Farnell, among other retailers. They can handle up to about 5A.

Much more significant currents can be handled using the devices in the parts list, which are not all that expensive but are unlikely to be available from your local shop (but kits are available). All the other parts in the design are bog-standard, and you will surely have them in your parts drawer or at your local shop.

## Design process

Between Ian's initial email with the circuit he uses in DC and low-power



**Figs.1(a) & (b): the two main ways to use the Discrete Ideal Bridge Rectifier. At the top, a centre-tapped transformer secondary winding is used to generate split (positive and negative) rails. Two separate secondaries can also be used if they are connected in series. The connections at right show how to use the same board to combine the outputs of two DC supplies (the solar panel and battery are just examples). OUT+ will be fed by whichever has a higher voltage.**

# Bridge Rectifier

- Generates split rails (positive and negative DC supplies) from a single centre-tapped transformer secondary (or two secondaries wired in series)
- It can also be used to combine two DC supplies (whichever has a higher voltage feeds the load)
- Maximum output voltage: ±40V or +80V (transformer applications), +40V (combining DC supplies)
- Maximum current: 10A RMS without heatsinking, more with heatsinking
- Typical voltage drop: <100mV input-to-output
- Typical dissipation: 1.7W @ 5A RMS, 6.8W @ 10A RMS

AC applications and the final design presented here, we exchanged many ideas, questions and refinements. Some requirements we decided on are:

☑ A low part count was important.

☑ The design had to 'just work' without tweaks.

☑ Reverse current when Mosfets switch on and off had to be minimal in all applications.

Many ideas were shared, and challenges were presented in every direction. In the process, the conceptual circuit grew to something larger and more complicated than was strictly necessary. It was at this stage that we tabled those design goals.

Ian was keen to keep the size of the board down, so we designed a through-hole version and an alternative that uses some SMDs to fit in tighter spaces. We realised that this would never be the size of a conventional bridge rectifier, so we just aimed to produce reasonably-sized boards that would likely fit into an existing chassis but that aren't too fiddly to build either.

The final design is vastly 'tighter' than the test board. I often build a prototype board that is purely functional and worry about improving the layout later, once I've proven it works.

In discussing what changes were warranted to Ian's concept, achieving a design that 'just worked' became important. That led to the introduction of constant current sources as loads in the design. It makes the operation largely independent of supply rail voltage and allows constructors to use the Ideal Bridge with 9-25V AC transformers without any changes.

We also changed the sense circuit to only switch on the Mosfet when the input is at a programmed voltage above the rectified output. This blocks reverse currents and allows it to be safely used for combining DC supplies, which might be very close in voltage at times.

The resulting circuit is simple and works well. We'll get to a couple of subtleties later in the description, once we've gone over its operating principle.

## Two versions

There are two PCBs for this project: an SMD version and a through-hole version. They use the same circuit. The SMD version is smaller than the through-hole version, which may be helpful in some circumstances. It doesn't use any tiny parts (the resistors are M3216/1206 and there are SOT23 transistors), so it isn't hard to assemble.

Both versions use the same TO-220 (through-hole) Mosfets. That is because it makes it easy to add flag heatsinks if necessary for your application. High-current SMD Mosfets are available, but they are trickier to heatsink if necessary and will take up more room than a TO-220 in this application.

## Design limitations

This circuit is suitable for rectifying the output of dual or split secondary transformers where the junction of the windings from the ground point for output capacitors, as shown in Fig.1(a).

This design will work if you have a transformer with a single secondary winding, but the switching could be noisy. ICs like the LT4320 used in the November 2024 designs switch the bottom Mosfets on for a full half-cycle to ensure clean switching. So, for that sort of application, we recommend you build one of the designs we published then (kits are available at siliconchip.au/Shop/?article=16043).

Regarding how much current the board can handle, P-channel Mosfets typically have a higher $R_{DS(ON)}$ figure than N-channel Mosfets. This means that the positive-rail Mosfets will be the limiting factor in how much current can be drawn due to their voltage drop and consequent power dissipation.

We have avoided the complexity of a gate drive boost circuit there. Using one would have allowed us to use four identical N-channel Mosfets, but we didn't think that was worth the extra parts and possibly new failure modes.

Up to about 10A, the Mosfets will not require heatsinking, although it wouldn't hurt to add small flag heatsinks above 5A. Above 10A, you must add a substantial flag heatsink on each Mosfet. Decent flag heatsinks should let it handle at least 15A. Beyond that, you might need a more serious cooling solution, like forced airflow over heatsinks.

## Circuit details

The circuit is shown in Fig.2. Unlike the previous Ideal Bridge Rectifier, this circuit can have its inputs connected across a single secondary or a pair of series-connected secondaries to generate split supply rails. In those cases, the secondary winding's centre tap does not connect to this circuit. Instead, it connects to the output capacitor bank ground and the load's ground, as shown in Fig.1(a).

So that it can produce split rails, it contains two similar sections stacked on top of each other. They would be identical except that they have opposite polarities to handle current flowing in opposite directions. The upper section uses two P-channel Mosfets, four PNP bipolar junction transistors (BJTs) and two NPN BJTs. The lower section has two N-channel Mosfets, four NPN BJTs and two PNP BJTs.

Each of the four sections senses the input AC voltage at one terminal. When it is about 34mV greater in magnitude than the output voltage (higher than the positive rail or lower than the negative rail), the corresponding Mosfet is switched on by driving its gate with an appropriate voltage.

We only want the Mosfet on when the input exceeds the output by a small margin to ensure that the Mosfet is off when these voltages are equal and that there is no chance the Mosfet is on as the input voltage magnitude drops below the output.

If that were to occur, current would reverse and flow from the capacitor bank through the transformer, creating current spikes and a great deal of electrical noise, plus possibly overheating the Mosfets.

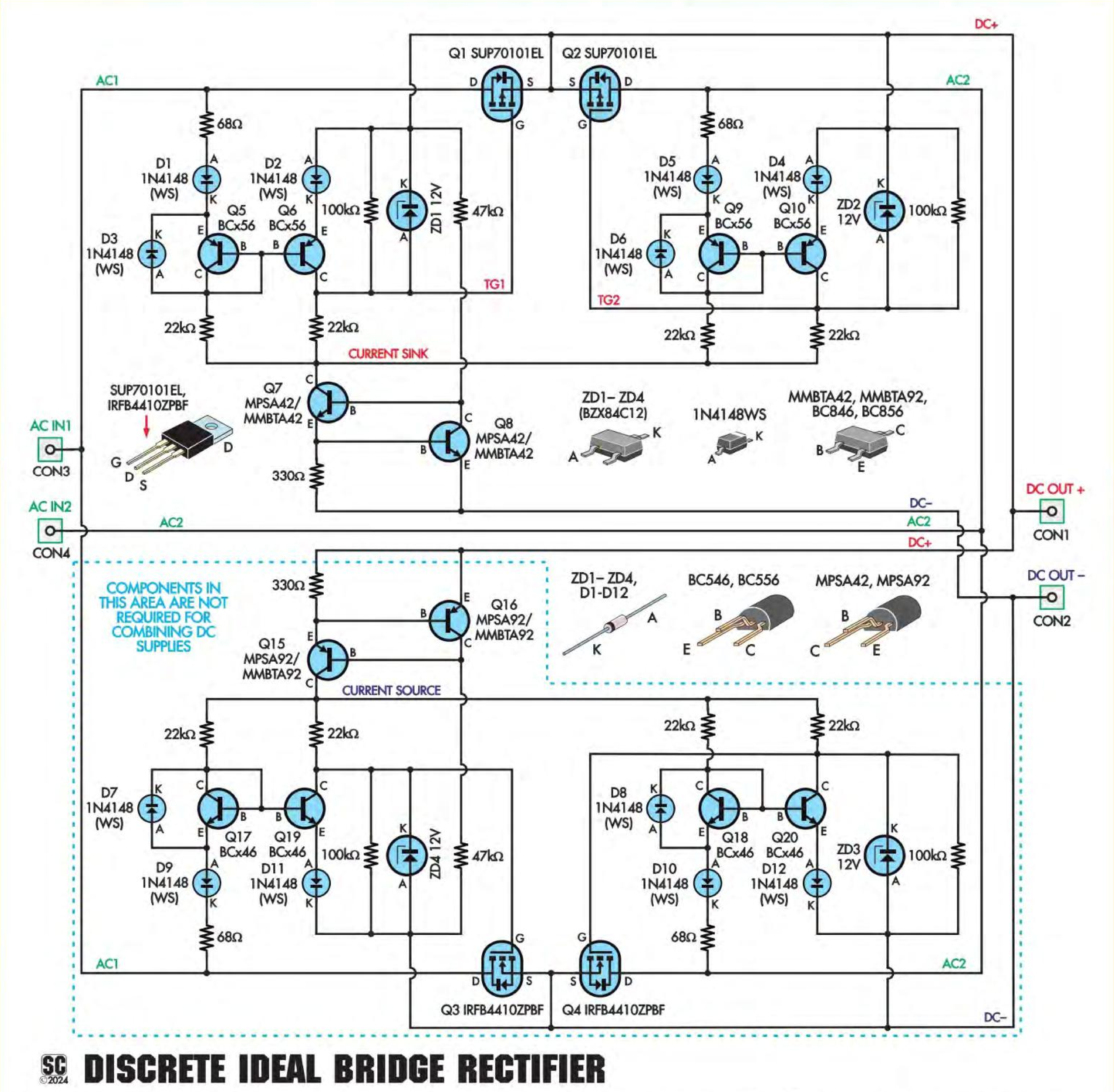
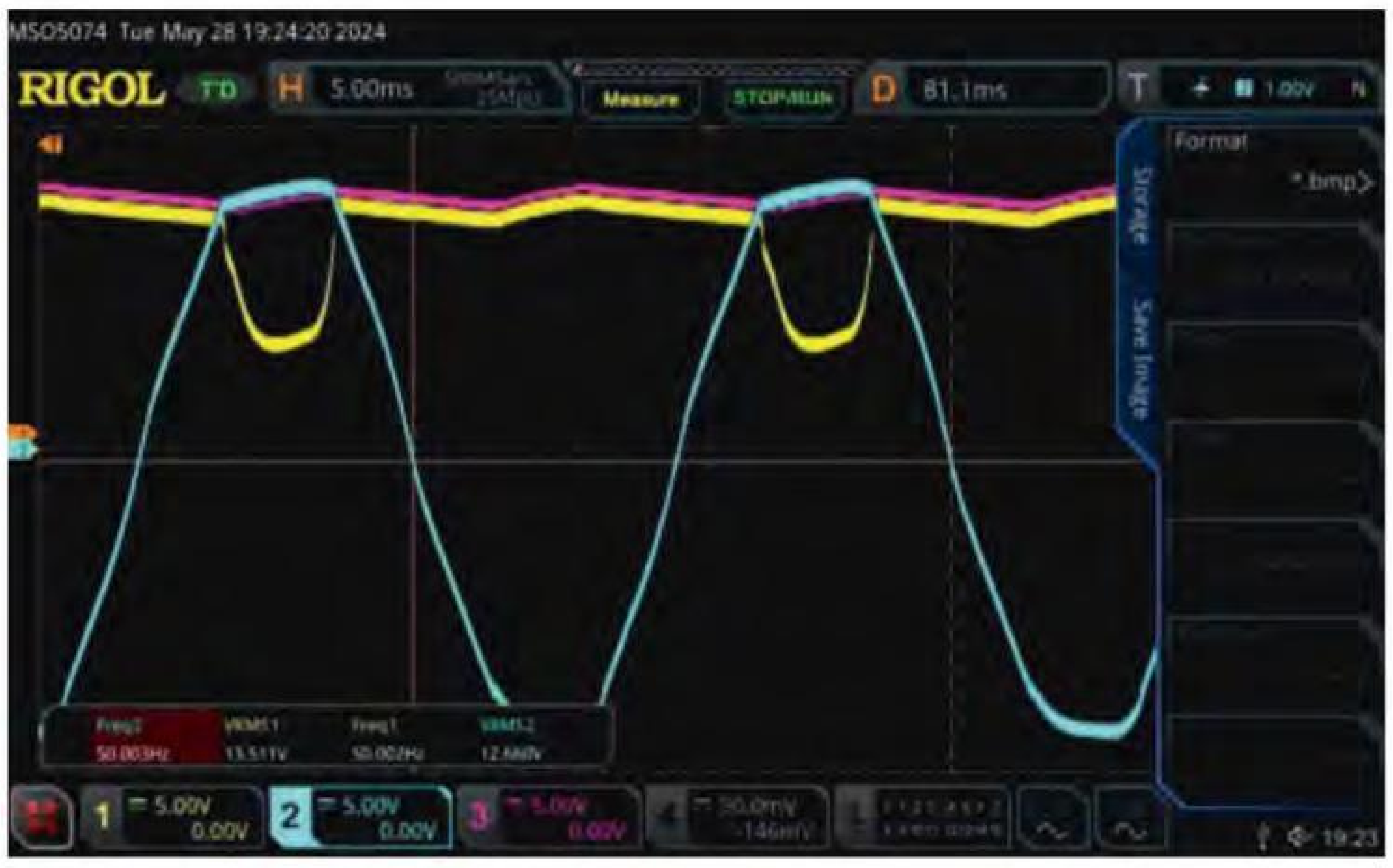
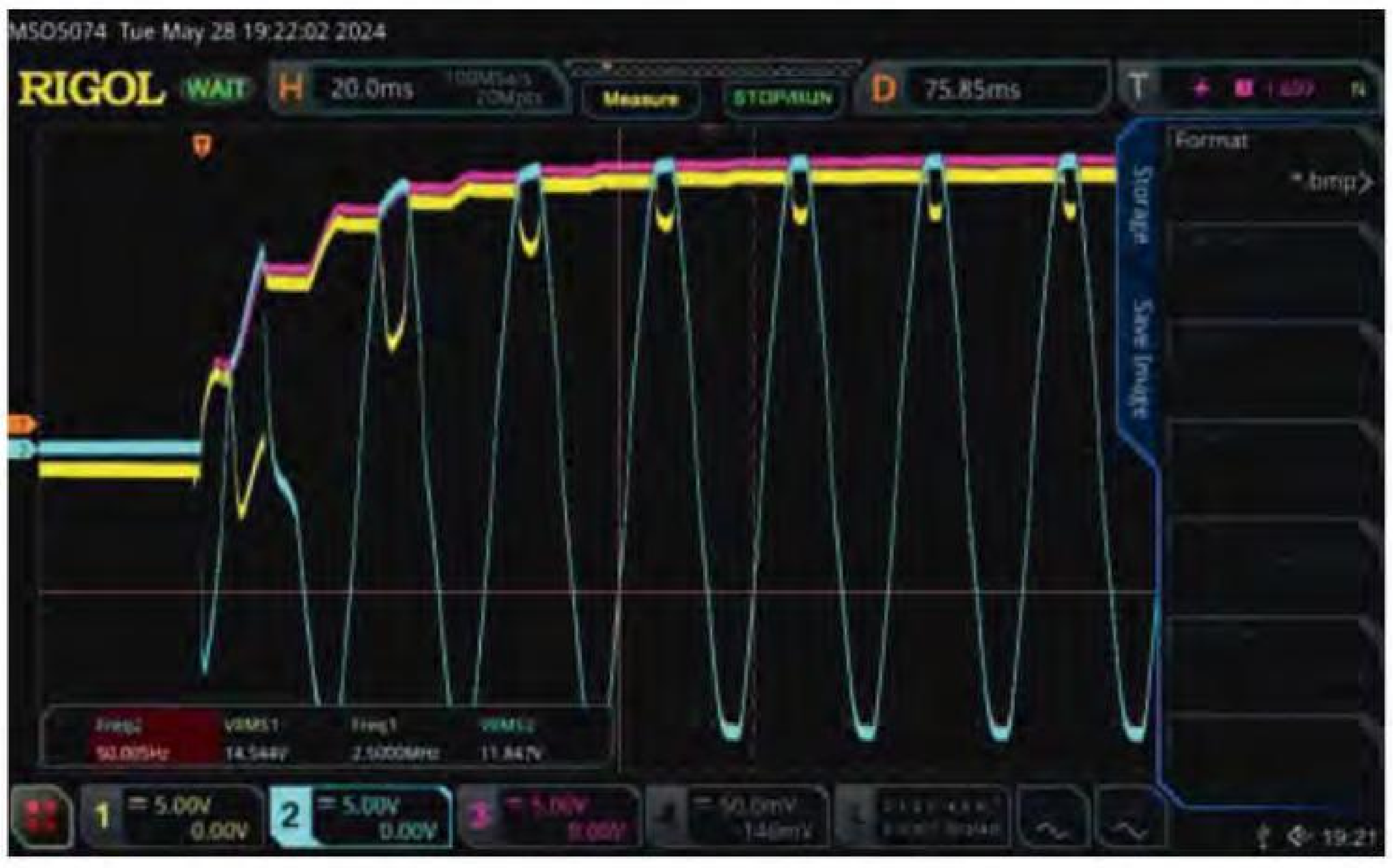

DC+
Q1 SUP70101EL
Q2 SUP70101EL
AC1
AC2
68Ω
D1 1N4148 (WS)
D2 1N4148 (WS)
D3 1N4148 (WS)
Q5 BCx56
Q6 BCx56
100kΩ
ZD1 12V
47kΩ
TG1
TG2
D5 1N4148 (WS)
D4 1N4148 (WS)
D6 1N4148 (WS)
Q9 BCx56
Q10 BCx56
ZD2 12V
22kΩ
CURRENT SINK
Q7 MPSA42/ MMBTA42
Q8 MPSA42/ MMBTA42
330Ω
SUP70101EL, IRFB4410ZPBF
G D S
ZD1– ZD4 (BZX84C12)
1N4148WS
MMBTA42, MMBTA92, BC846, BC856
AC IN1
CON3
AC IN2
CON4
DC–
DC OUT +
CON1
DC OUT –
CON2
COMPONENTS IN THIS AREA ARE NOT REQUIRED FOR COMBINING DC SUPPLIES
Q16 MPSA92/ MMBTA92
Q15 MPSA92/ MMBTA92
CURRENT SOURCE
ZD1– ZD4, D1-D12
BC546, BC556
MPSA42, MPSA92
D7 1N4148 (WS)
Q17 BCx46
Q19 BCx46
D9 1N4148 (WS)
D11 1N4148 (WS)
ZD4 12V
D8 1N4148 (WS)
Q18 BCx46
Q20 BCx46
D10 1N4148 (WS)
D12 1N4148 (WS)
ZD3 12V
Q3 IRFB4410ZPBF
Q4 IRFB4410ZPBF
SC ©2024
DISCRETE IDEAL BRIDGE RECTIFIER

**Fig.2: the Ideal Bridge Rectifier circuit comprises two identical sections at the top to deliver current to the DC OUT+ terminal, with two more sections below to handle current flow through the DC OUT– terminal. The lower sections are 'mirror images' of the upper sections, with components of opposite polarity (NPN transistors instead of PNP etc). The circuit is the same for the TH and SMD versions; the alternative devices are direct equivalents except for their packages.**



**Scope 1: an oscilloscope grab of the Ideal Bridge in operation, showing rectification of the voltage at the AC1 terminal. The pink trace is the output voltage at 5A, cyan is the AC input voltage, and yellow is the gate drive for Mosfet Q1, which peaks at about -8V.**



**Scope 2: the Discrete Ideal Bridge starting into two 35,000µF capacitor banks. This is a pretty brutal thing to do to any bridge. Usually, you would use a soft-start circuit to keep the initial current surge under control. Still, the Bridge survived it!**

As the four separate sections all work the same way, let's concentrate on the one shown in the upper-left corner of Fig.2. The voltage sensing circuit comprises diodes D1 and D2 plus PNP transistors Q5 and Q6. Q5 acts as a diode, since its base and collector are joined.

Ignoring the 68Ω resistor for now, with a constant current flowing through these transistors, both will conduct if the AC input voltage at CON3 is the same as the DC output voltage at CON1. If Q6 is on, Mosfet Q1's gate voltage is high, and it is off. As the input voltage increases, Q6 switches off, so the gate of Mosfet Q1 is pulled low by its 22kΩ collector resistor – see Scope 1.

The 68Ω resistor is important as it alters how the comparator works. The total current through the two 22kΩ collector resistors is determined by a constant current sink comprising NPN transistors Q7 and Q8. On the positive cycle for the AC1 input, about 0.5mA is drawn through each of these resistors (as well as the matching pair for Q9 & Q10).

This 0.5mA flows through the transistor and diode pairs Q5/D1 and Q6/D2, which drop the voltage by about 1.2V, but on the AC input path, it also flows through the 68Ω resistor, dropping 34mV or so in the process.

This extra voltage drop means we draw more current from the base of Q6 than Q5 until the AC input is 34mV above the output voltage. Mosfet Q1 remains switched off until that condition is met. Once the input exceeds the output by 34mV, Q6 starts switching off and the Mosfet switches on. This charges the output capacitors until they get to 34mV below the input.

Essentially, the circuit contains a negative feedback loop, where Q5 and Q6 try to maintain a 34mV difference across the Mosfet by controlling its gate voltage. Without the 68Ω resistor, they would try to maintain 0V across the Mosfet, and due to various tolerances in the circuit, the Mosfet might be held on all the time, which is not what we want!

As a result, at lower load currents, we are not simply switching the Mosfet hard on and off; instead, it is operating in linear mode with a low voltage drop across it due to the negative feedback. Part of that voltage drop is a result of the $R_{DS(ON)}$ of the Mosfet, while part is from the gate voltage being moderated, which we can see in the oscilloscope screen grabs.

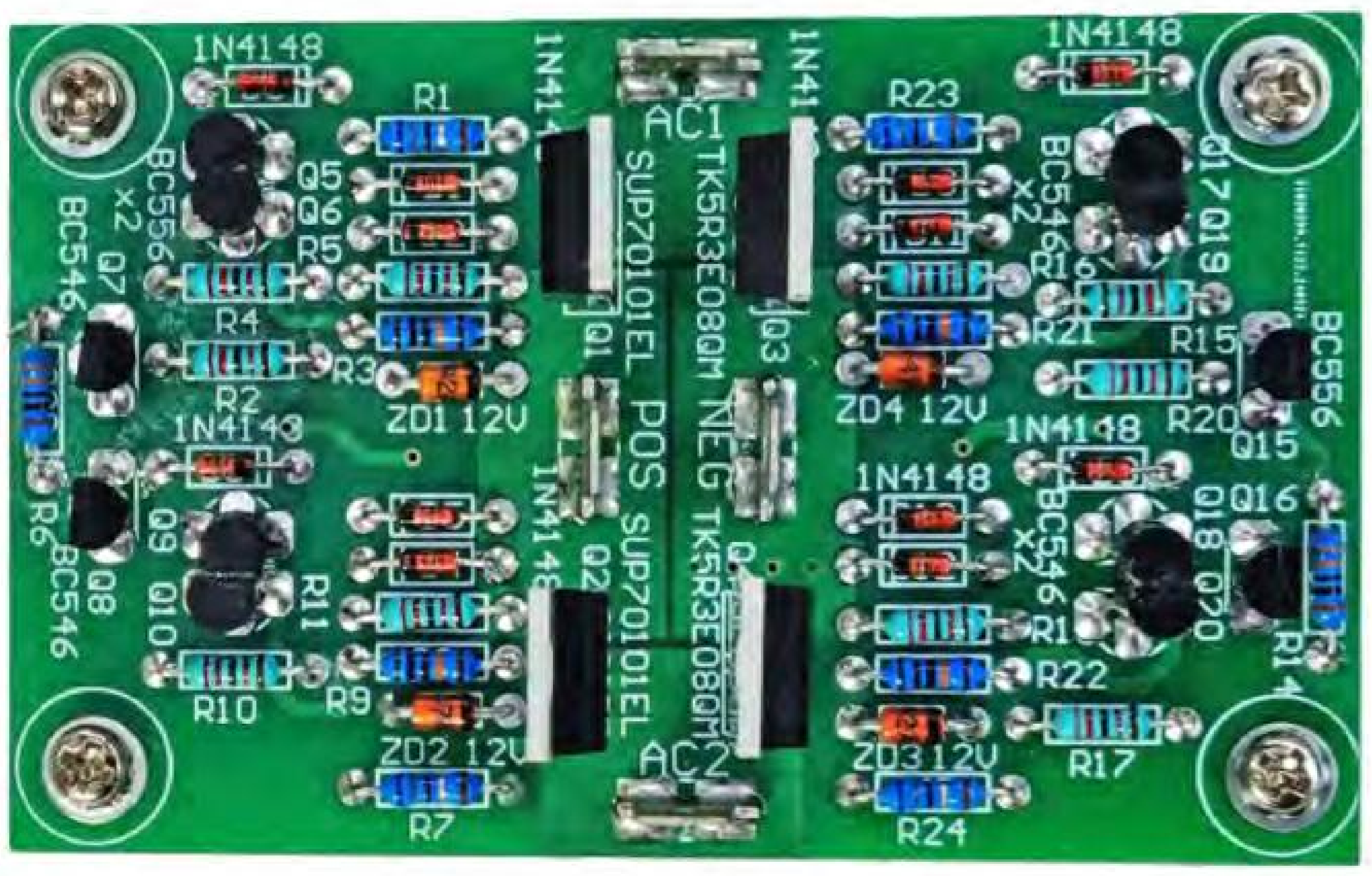
As the load current increases, we see the sense circuit driving the Mosfet harder, ie, its $V_{GS}$ increasing until it is 12V, at which point the gate protection zener diode (ZD1 in this case) conducts to prevent the Mosfet gate from being driven beyond its ratings.

If you look at the scope images (especially Scope 4), you will see that when drawing high currents, the circuit transitions from the linear feedback operation to driving the Mosfet fully on with 12V. This occurs because the voltage drop across the Mosfet exceeds 34mV due to its minimum $R_{DS(ON)}$.

As a result of the way we are driving the Mosfet, there is little value in utilising ultra-low $R_{DS(ON)}$ Mosfets in a dual-rail bridge. 10mΩ or so is fine. We felt this was the sweet spot at which the voltage drop across the Mosfets is defined by the feedback loop up to about 5-6A. Because of how Mosfets are made, P-channel Mosfets tend to have a higher $R_{DS(ON)}$.

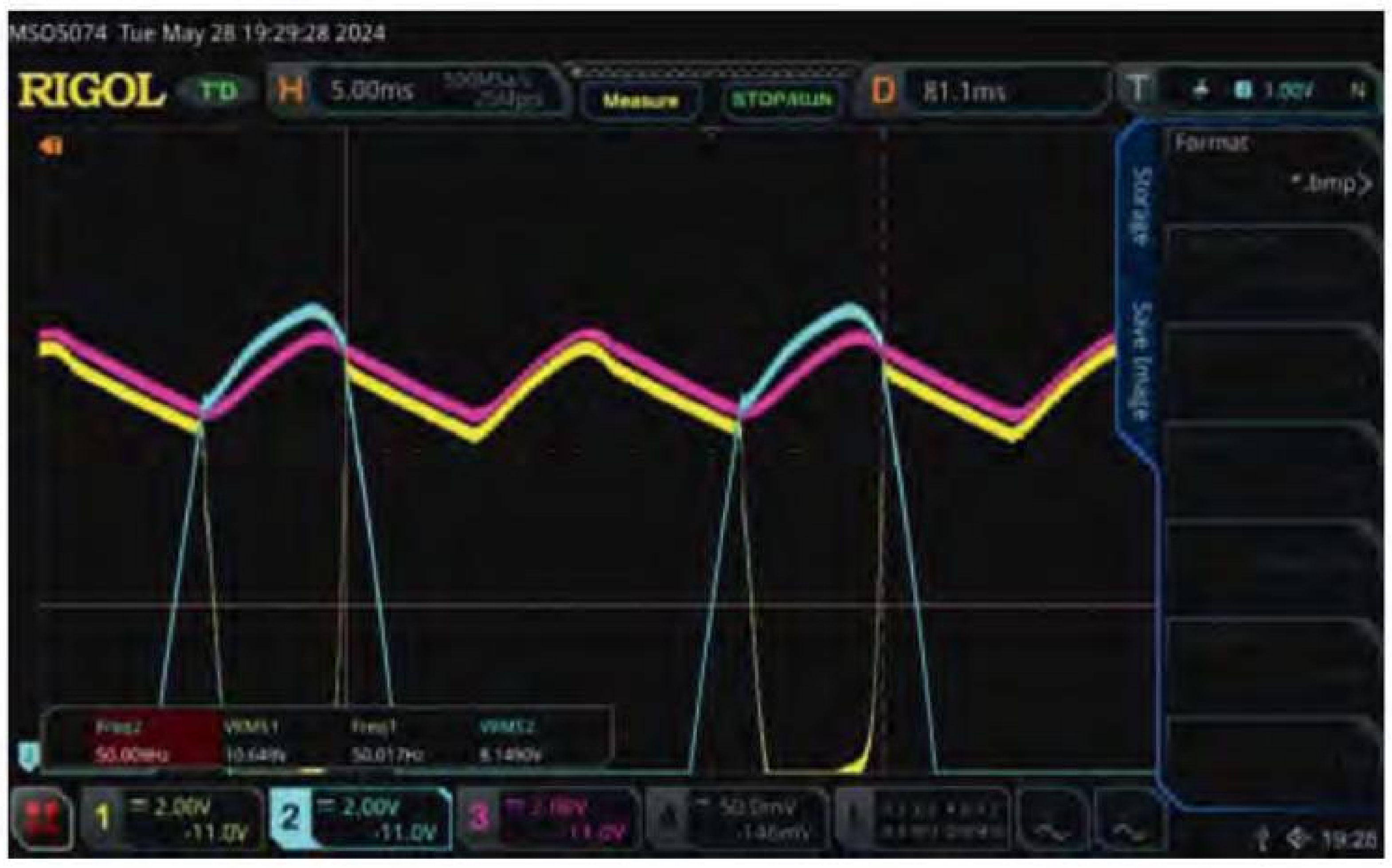
The constant current sink based around Q7 & Q8 is a standard two-transistor current source/sink configuration. We could have tied this to the output ground and reduced the dissipation in transistor Q7, but we chose to tie it to the negative output rail for the positive rail comparators and positive rail for the negative comparators.

This is because it gives maximum gate drive to the Mosfets for low-voltage operation, especially during startup when massive currents are often drawn for charging capacitor banks. This

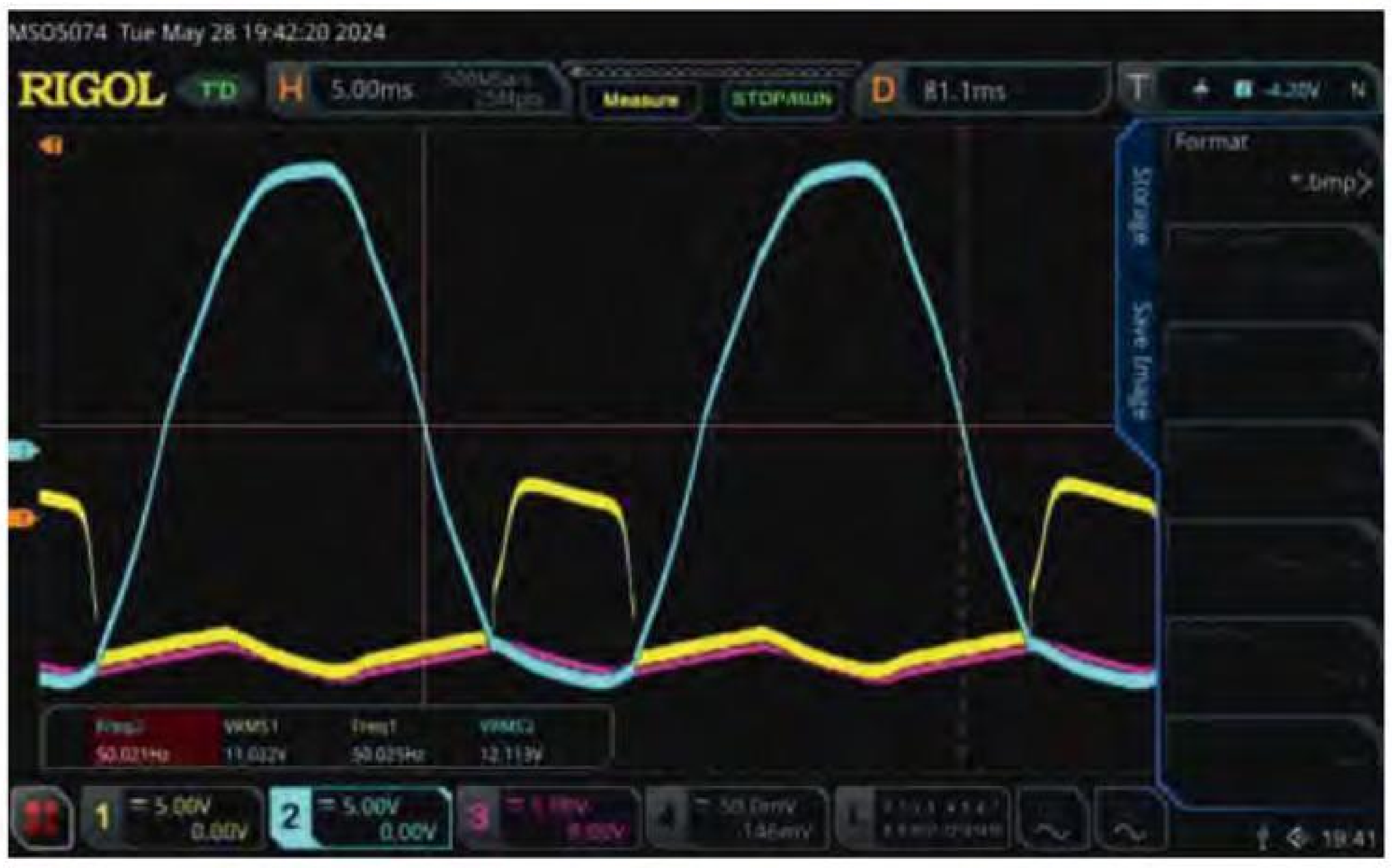


**This version of the Ideal Bridge Rectifier uses all through-hole components. Note the pairs of transistors joined face-to-face so they track thermally.**

**Scope 3: a close-up of the rectified output. Again, pink is the output, cyan is the AC input, and yellow is the gate drive. This neatly shows the Mosfet switching as the AC input voltage slightly exceeds the DC output voltage.**



**Scope 4: the negative rail behaviour, which is a 'mirror image' of the positive rail. Driving the rectified 12V AC into a 1Ω load is clearly giving the transformer a workout, as seen by the flattened top and bottom of the cyan waveform. Under these conditions, it would be advisable to mount a flag heatsink to each Mosfet as they individually dissipate about 1.5W.**



reduces dissipation in the Mosfets during these high-stress phases of operation. It also has the benefit of the PCB not needing a GND connection.

If you look at Scope 2, which shows the startup behaviour, the Mosfet has over 10V of gate drive in the first cycle of operation. One benefit of using a constant current source/sink is that the circuit's behaviour is mostly independent of the operating voltage, as long as it's above the minimum threshold required to bias on the Mosfets.

The 22kΩ resistors in the circuit allow one current source/sink to drive the sense amplifiers for both input rails. The actual value of these resistors is not that important, although we don't want a large voltage drop across them so that we can use the Ideal Bridge at modest AC input voltages.

For the PCB layout we need to consider the thermal characteristics of D1, D2, Q5 and Q6 (the sense amplifier). Silicon diodes have a -2.1mV/°C thermal coefficient for their forward voltage drop, so for every 1°C increase in temperature of a diode junction, its forward voltage falls by 2.1mV.

This means that if one diode is hotter than the other, we will get an error in the switching voltage. A similar effect is seen with the base-emitter voltages of Q5 and Q6. For this reason, we have placed the diodes right next to one another, and placed the transistors so they can be glued together. This will ensure our switching margins are stable even as the board heats and cools during use.

The 'sense' transistors (Q5 & Q6, Q9 & Q10 etc) only ever have 12V across their collector-emitter junctions, so we have specified standard BC546-9 or 556-9 devices (or their SMD equivalents, BC8xx).

However, the current source/sink transistors will have the full dual rail voltage across them, which could be up to ±40V or 80V total. Therefore, we have specified MPSA42/92 transistors for these (or the SMD equivalents, MMBTAx2).

These standard high-voltage, low-power devices are available from all the larger online suppliers. If you have ±25V or lower voltage rails, you could use BC546/556/846/856 transistors there instead. It is important to consider that the BC546/BC556 have the opposite pinout to the MPSA42/92 transitors, so you would need to install them backwards if you do this.

Luckily, for the SMD transistors, the BC846/856 series SMD pinouts are the same as the MMBTA42/92 pinouts, so they are a direct swap for applications below ±25V.

Note that the 47kΩ resistor values were chosen to allow operation from low voltages to about ±40V at the output. At the upper limit, the 47kΩ resistors will dissipate 130mW each. While that is well within the ratings of a ¼W resistor, we have specified ½W resistors just to be safe.

If you will only use this bridge at the higher end of its voltage range, you could increase those resistor values slightly to, say, 68kΩ. That will reduce their dissipation to a maximum of

| Application | Max current | Max voltage | N-channel | P-channel | Source/comments |
|---|---|---|---|---|---|
| Low-Current Full Bridge | 2-3A *no heatsink* | 40V | IRF540 | IRF9540 | Farnell 8648298 |
| High-Current Full Bridge | 10A *no heatsink* | ±40V | IRFB4410ZPBF | SUP70101EL-GE3 | Mouser, DigiKey & Silicon Chip kit |
| | | | IRF135B203 | IXTP76P10T | |
| | | ±30V | As above | SUP90P06 | |
| | | | | IXTP96P085T | |
| DC Combining | 5A *no heatsink* | 12-24V | Not required | IRF9540 | Farnell 8648620<br>100mV/A drop |
| DC Combining | 10A *no heatsink* | 12-24V | Not required | SUP90P06-09L-E3 | Mouser & DigiKey<br>7.4mV/A drop |
| | | | | SUP70101EL-GE3 | Mouser & DigiKey<br>11.4mV/A drop |
| | | | | IRF4905 | Mouser & DigiKey |

**Table 1 – examples of suitable Mosfets**

94mW, so ¼W resistors should be fine. You could also lower their values for low-voltage applications, although that shouldn't be necessary.

### Startup behaviour

Scope 2 shows the circuit starting up when AC power is first applied. On that first cycle, the AC input blue trace goes negative. This charges the negative capacitor to about 5V, although we don't have a plot of the negative rail here – we know that the negative and positive rails will be about the same.

The Mosfet body diode conducts on this cycle in the absence of voltage at the Mosfet gate (due to the low initial voltage). Once there are a few volts on the output rails, the constant current source/sink and BJT-based voltage sense circuits kick in. By the time we are into the first positive excursion of the AC1 input in cyan, we can see the gate drive pulling the gate low (in yellow), having already charged the large capacitor bank enough in the first cycle.

Indeed, the gate voltage on that P-channel Mosfet goes below 0V, being pulled toward the negative rail, and we see a full 12V on that P-channel Mosfet gate in the first real cycle of operation. This shows the benefit of connecting the current source/sink to the opposite rail rather than ground.

I love the simplicity of circuits like this, which squeeze more out of a handful of components than seems reasonable. I also like going back to basics and using BJTs in the current sink and sense amplifier.

### PCB layout

We touched on some PCB layout considerations earlier. There are a few aspects of the PCB design that are very important:

- The layout of the current sense amplifier with its two transistors, two 1N4148 diodes and 68Ω resistor is kept very tight as it must accurately sense small voltages with relatively low bias currents.
- The sense transistor pairs, like Q5 and Q6, are face-to-face, so you can super glue these together to keep them as tightly thermally coupled as possible (or add a smear of thermal paste between them). On the SMD version, these parts are tight against one another.
- The pairs of 1N4148 diodes (D1 & D2) are right next to one another, so they stay at similar temperatures.
- The path from the AC inputs through the Mosfets and to the DC outputs is kept as short as possible and uses large copper fills to maximise the current carrying capacity of the PCB. PCBs do not have a fixed 'current rating', but we must ensure that the voltage drop and heating in the tracks is reasonable at any current likely to be drawn. At the AC1 input, which has the thinnest connection to the Mosfet, we have parallel copper on the top and bottom layers of the PCB.

## Parts List – Discrete Ideal Bridge Rectifier

4 6.3mm pitch PCB-mount vertical spade connectors (CON1-CON4)
2 SUP70101EL 100V 120A P-channel Mosfets, TO-220 (Q1, Q2)
2 IRFB4410ZPBF 100V 97A N-channel Mosfets, TO-220 (Q3, Q4)

**Resistors** *(1% ¼W axial – TH version | 1% ¼W M3216/1206 – SMD version)*
4 100kΩ 2 47kΩ 0.5/0.6W (5% OK) 8 22kΩ 2 330Ω 4 68Ω

### Through-hole version

1 double-sided PCB coded 18108241, 87.5 × 45.5mm
4 BC556/7/8/9 100mA PNP transistors, TO-92 (Q5-Q6, Q9-Q10)
2 MPSA42 300V 500mA NPN transistors, TO-92 (Q7, Q8)
2 MPSA92 300V 500mA PNP transistors, TO-92 (Q15, Q16)
4 BC546/7/8/9 100mA NPN transistors, TO-92 (Q17-Q20)
4 12V 0.4W zener diodes, DO-35 (ZD1-ZD4)
12 1N4148 75V 200mA diodes, DO-35 (D1-D12)

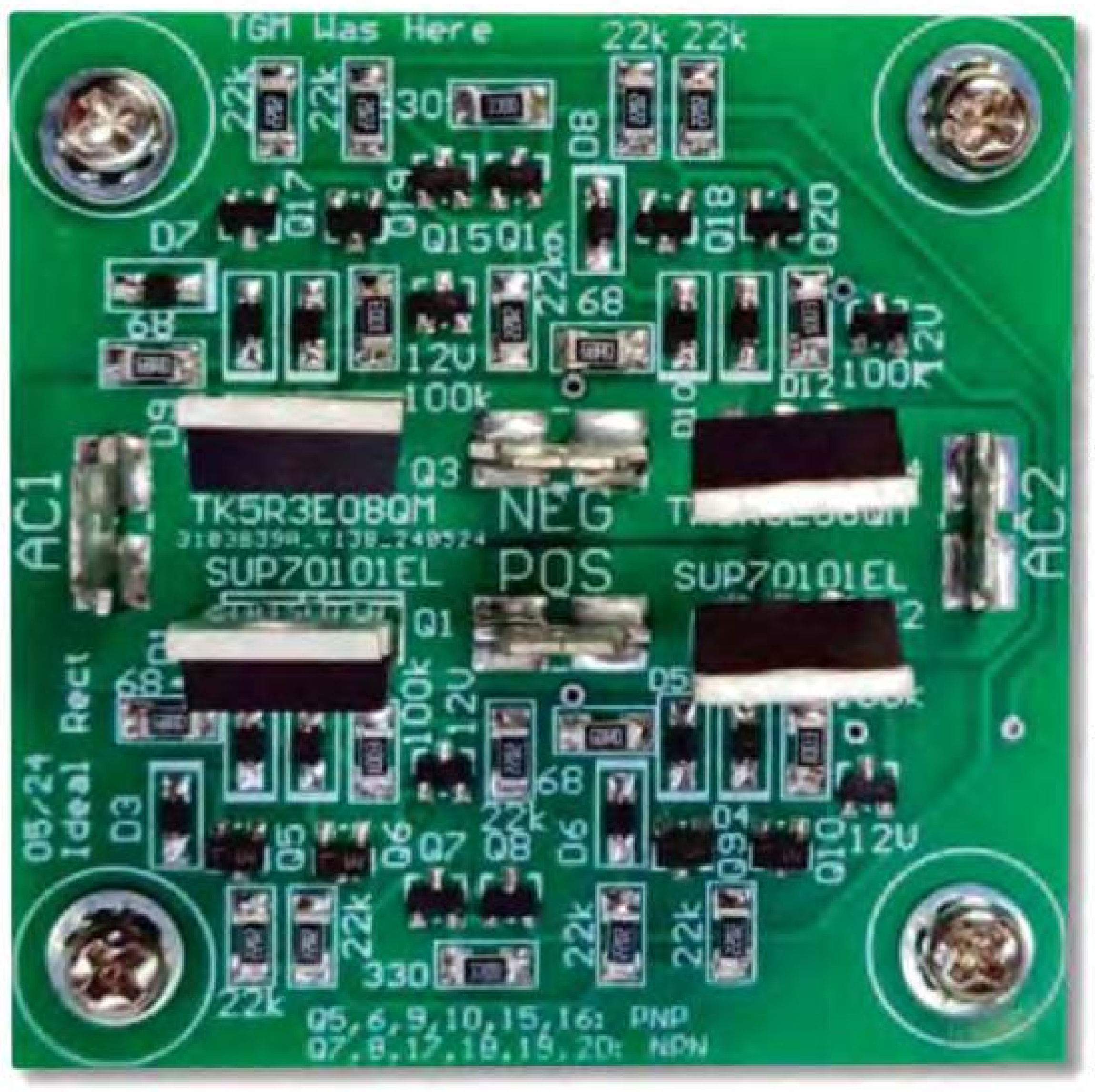
### SMD version

1 double-sided PCB coded 18108242, 54.5 × 54.5mm
4 BC856/7/8/9 100mA PNP transistors, SOT-23 (Q5-Q6, Q9-Q10)
2 MMBTA42 300V 500mA NPN transistors, SOT-23 (Q7, Q8)
2 MMBTA92 300V 500mA PNP transistors, SOT-23 (Q15, Q16)
4 BC846/7/8/9 100mA NPN transistors, SOT-23 (Q17-Q20)
4 12V ¼W zener diodes, SOT-23 (ZD1-ZD4) **[BZX84C12]**
12 1N4148WS 75V 150mA diodes, SOD-323 (D1-D12)

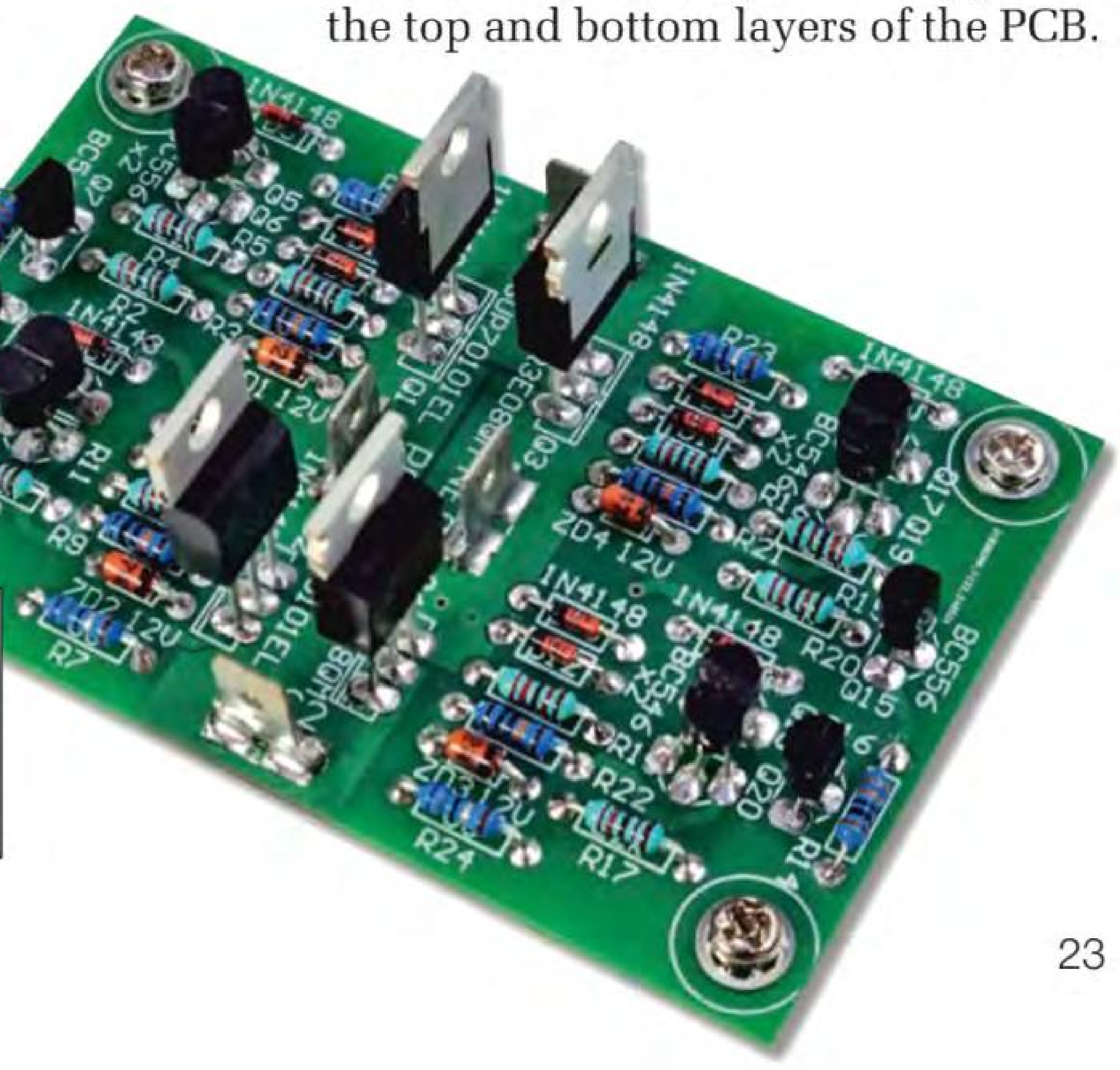
*For combining DC supplies, halve the numbers of all components except the PCB and spade connectors.*

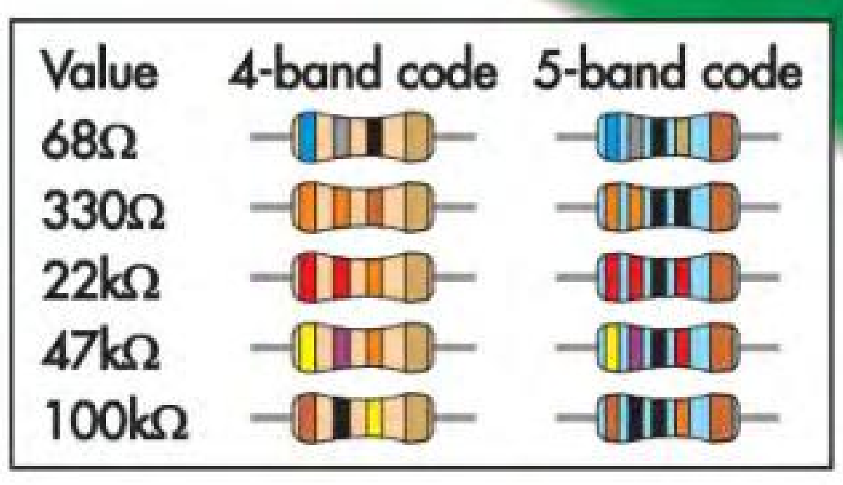
- **TH version kit (SC6987, ~£15)**
- **SMD version kit (SC6988, ~£14)**

*Both kits from Silicon Chip include the PCB and everything that mounts on it.*



**The SMD version of the Discrete Ideal Bridge Rectifier is 54.5 × 54.5mm, while the through-hole only is a bit larger at 45.5 × 87.5mm (not shown to scale).**





Value
4-band code
5-band code
68Ω
330Ω
22kΩ
47kΩ
100kΩ

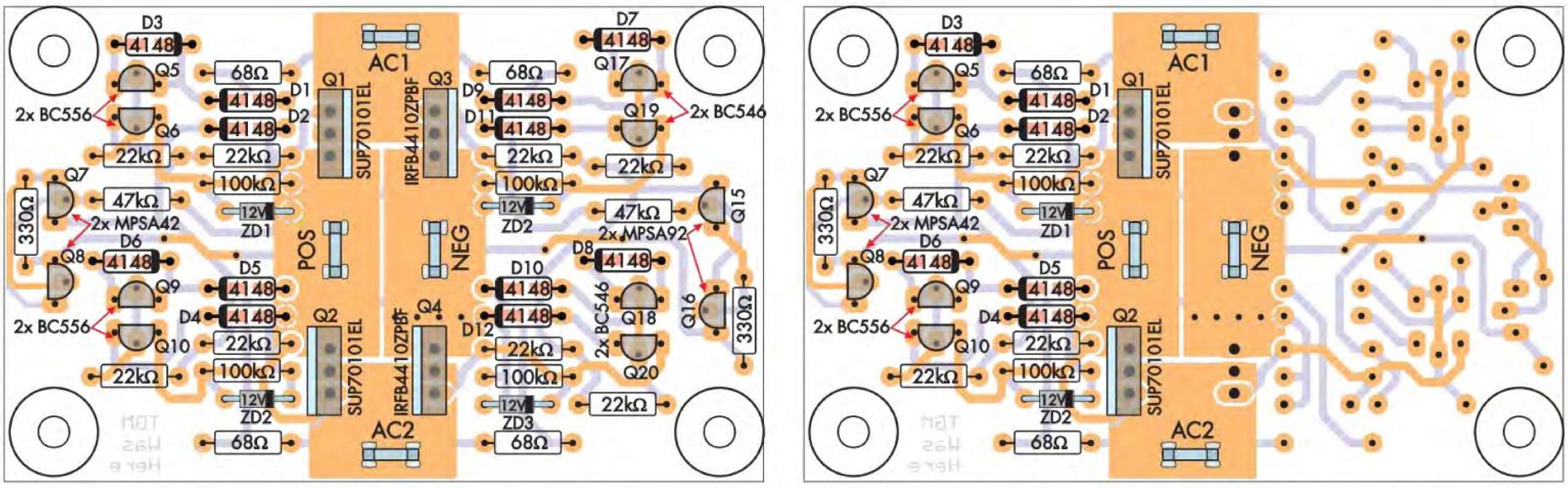

D3
4148
Q5
68Ω
2x BC556
Q6
22kΩ
100kΩ
Q7
330Ω
47kΩ
2x MPSA42
D6
Q8
Q9
Q10
D1
D2
D4
D5
12V
ZD1
ZD2
Q1
SUP70101EL
Q2
POS
AC1
AC2
NEG
Q3
IRFB4410ZPBF
Q4
D7
D8
D9
D10
D11
D12
Q17
Q19
2x BC546
ZD2
ZD3
Q15
Q16
2x MPSA92
Q18
Q20
TGM
Was
Here

**Fig.3(a) & (b): the full-populated through-hole version of the PCB (left) and the reduced version for combining DC supplies only (right). The full version can also be used to combine DC supplies. Watch the diode and Mosfet orientations, and remember that Q7/Q8 and Q15/Q16 need to be reversed if you are using BC546/BC556 transistors instead for lower voltage applications, compared to what's shown here.**

## Mosfet selection

We have included 100V low-$R_{DS(ON)}$ Mosfets in the parts list. They only cost a few dollars each and work well. If selecting alternative Mosfets, look for a voltage rating well above the rail voltage you want; we feel that 80-100V is about right. Select an $R_{DS(ON)}$ of 10mΩ or less.

The P-channel Mosfet will usually have a higher $R_{DS(ON)}$; there is little point in selecting N-channel Mosfets with a significantly lower on-resistance than the P-channel devices you will be using.

For lower currents, you can get away with less expensive Mosfets. Even though the savings in dissipation won't be as great, the reduction in voltage loss can still make this design very beneficial in lower-current designs.

For example, we used IRF540/IRF9540 Mosfets from Altronics in some tests, and it was fine up to about 3A, still giving a much lower voltage drop than a conventional bridge. Table 1 includes some advice on Mosfet selection.

## Construction

The through-hole version is built on a double-sided PCB coded 18108241 that measures 54.5 × 87.5mm, while the SMD version is coded 18108242 and is a bit smaller at 54.5 × 54.5mm. For the former, refer to the Fig.3(a) PCB overlay diagram, while Fig.4(a) is the overlay for the SMD version.

The smallest SMD parts are the SOT-23 transistors and SOD-323 diodes. These are large enough that they are not too challenging if you have a desk magnifier and a reasonably good soldering iron.

If you are using it to combine solar panels or DC power sources, you can leave off all the negative rail parts, shown in a dashed box in Fig.1. These versions are shown in the alternative overlay diagrams, Figs.3(b) & 4(b).

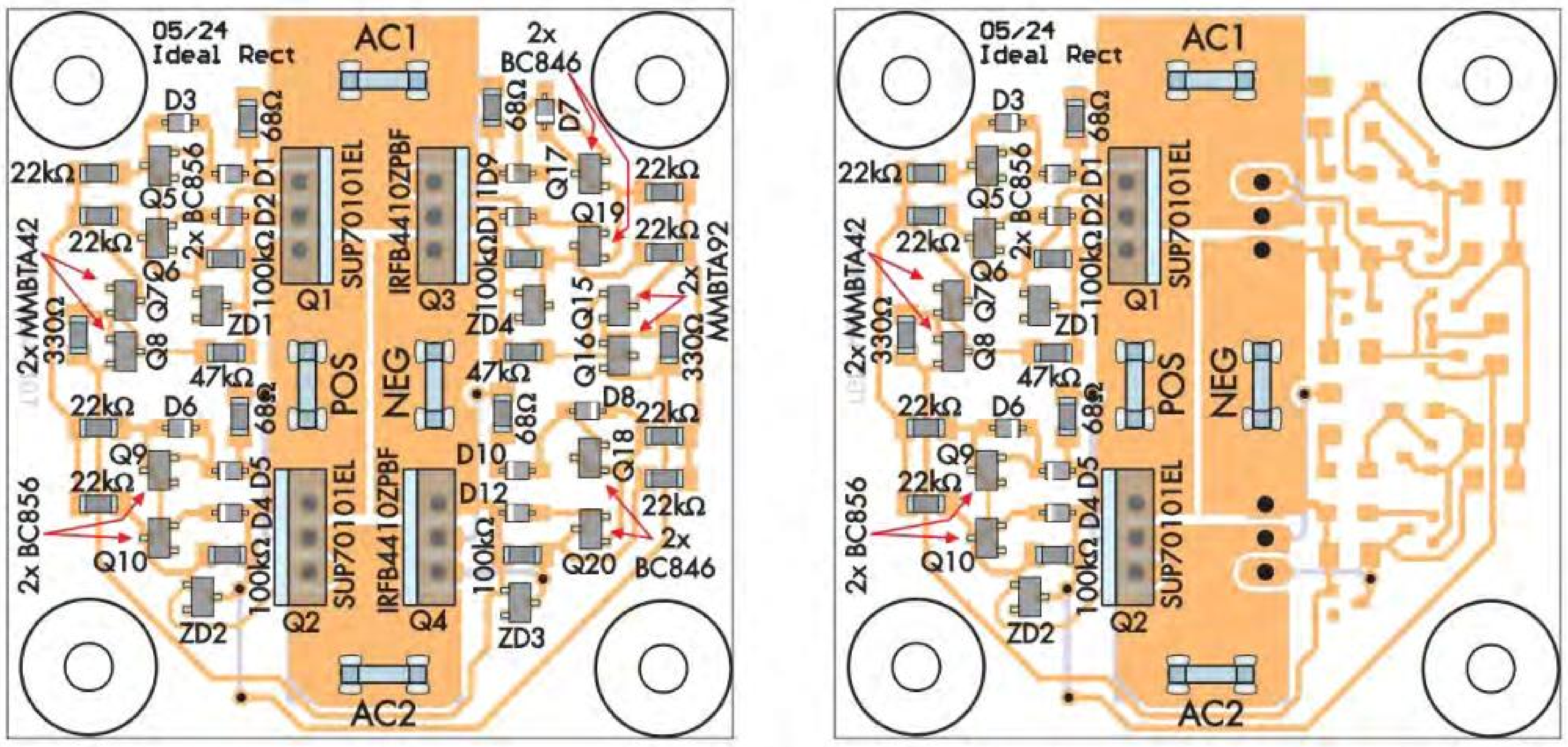
Start by fitting all the resistors. Follow with the diodes, making sure you orientate them correctly, with the cathode stripes facing as in the relevant PCB overlay diagram. We found that for the SOD-323 SMD diodes we got, it was tough to tell which end was the cathode. If unsure, use a magnifier or a DMM set on diode test mode.

Next, solder the signal transistors in place. As mentioned earlier, if you are using this at low voltages only, you can use all BC546/556/846/856 transistors throughout. If you do this, remember that the through-hole devices for Q7, Q8, Q15 & Q16 must be rotated by 180°, as the MPSA42/92 types have a different pinout.

Mount the 12V zener diodes next. The SMD SOT-23 parts are small and in the same packages as the bipolar transistors, so make sure you don't mix them up. Place them with tweezers and tack one leg, allowing you to adjust it (if necessary) by reheating the initial joint before soldering the remaining leads.

Fit the power Mosfets next. Watch the layout here, as they face in alternate directions on the board to optimise the track layout. Also, don't get the two different types mixed up. Tack one leg of each and fiddle them so they are neatly aligned and the same height, then solder the remaining leads.

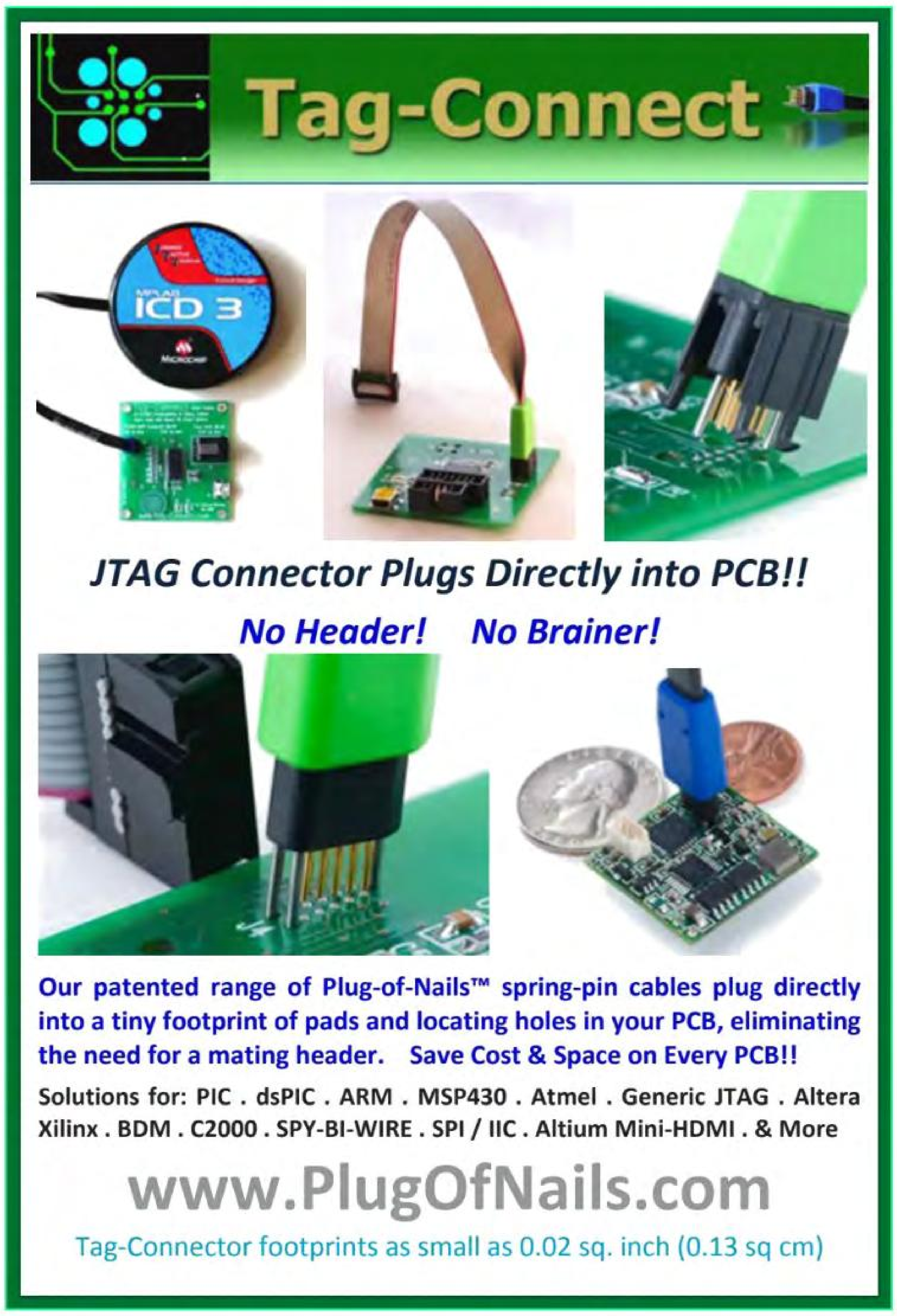
Finally, mount the 6mm connectors. You could solder wires directly to the board, but we reckon using crimp spade lugs is much neater.


05/24
Ideal Rect
AC1
AC2
D3
22kΩ
68Ω
2x BC846
2x MMBTA42
330Ω
Q5
Q6
Q7
Q8
Q9
Q10
2x BC856
100kΩ
D1
D2
D4
D5
D6
D7
D8
D9
D10
D11
D12
ZD1
ZD2
ZD3
ZD4
47kΩ
Q1
Q2
Q3
Q4
SUP70101EL
IRFB4410ZPBF
POS
NEG
Q15
Q16
Q17
Q18
Q19
Q20
2x MMBTA92

**Figs.4(a) & (b): the SMD versions of the PCB, with the full version on the left and the DC combining version only on the right. If substituting BC846/BC856 transistors for the MMBTA types, you don't need to change how they are fitted to the board. Only diodes D1-D12 and the Mosfets could be easily installed backwards, so ensure they aren't.**

## Testing

We suggest testing the board in two


Tag-Connect

JTAG Connector Plugs Directly into PCB!!

No Header! No Brainer!

Our patented range of Plug-of-Nails™ spring-pin cables plug directly into a tiny footprint of pads and locating holes in your PCB, eliminating the need for a mating header. Save Cost & Space on Every PCB!!

Solutions for: PIC . dsPIC . ARM . MSP430 . Atmel . Generic JTAG . Altera Xilinx . BDM . C2000 . SPY-BI-WIRE . SPI / IIC . Altium Mini-HDMI . & More

www.PlugOfNails.com

Tag-Connector footprints as small as 0.02 sq. inch (0.13 sq cm)

ENIGMA
COMPONENT SHOP

www.enigma-shop.com

Electronic Components for the Hobbyist!

Cable & Wire • Capacitors • Connectors • Filters
Kits & Modules • Hardware & Cooling
and much more...

READY TO BE REWARDED?

Join our loyalty scheme and **earn points** on every purchase you make which can be **used for discounts** in the future.

Read our reviews on Trustpilot

halves. The following steps test the two positive sections.

**1.** Connect the Bridge outputs to an electrolytic capacitor of at least 470µF. Make sure you get the polarity correct.

**2.** Connect the negative of a 12-24V power supply to the negative of your capacitor and the positive to either of the AC inputs. If you can set a current limit, set it to a few hundred milliamps.

**3.** Switch on the supply and check that the capacitor charges up to the input voltage.

**4.** Put a 100Ω 1W resistor (or similar) across the capacitor and check that the voltage across it does not droop significantly (no more than 100mV).

This verifies that the appropriate Mosfets are on; otherwise, the voltage would drop by 600mV or more. It also confirms there are no catastrophic shorts, or you would get smoke.

Now test the other AC input using the same method.

If you run into trouble in either case, go through the following checklist below:

**1.** Is your power supply going into current limiting? Use a multimeter to check for the expected voltage at the AC input.

**2.** Are your Mosfets the right way around?

**3.** Check that the diodes are all orientated correctly. If any are wrong, the Rectifier will not work.

**4.** Check your soldering and look for solder bridges.

**5.** Check that the current sink and source work by measuring the voltage between the base and emitter pins of Q8 and Q16. The reading should be close to 0.6V in both cases. Also check for a ~600mV $V_{BE}$ on Q7 and Q15. If the readings are low, check that the associated 47kΩ resistors are OK.

**6.** Check the voltage across the zener diodes. Are they the right way around? If the capacitor bank is charged up and there is no load resistor, the voltage across them should be low, while you should get a reading of several volts with the 100Ω resistor across the capacitor.

**7.** If the behaviour is correct for one AC input of the Bridge but not the other, check the circuitry around the misbehaving input and compare voltages to the other half.

**8.** If both inputs don't work, you have a systematic problem since they are essentially independent.

Having tested it with one polarity, switch off the supply and connect its positive output to DC OUT+ on the Bridge and the negative of your power supply to one of the input terminals. You should see the capacitor charge up to the input voltage again. Proceed with testing in this configuration as above.

## Using it

Once installed, it will pretty well look after itself. Refer to Figs.1(a) & (b) to see how the connections should be made. If you expect to draw continuous high currents from the power supply, you will probably want to put some flag heatsinks on the Mosfets. Aside from that, you should find that it just works.

Remember that you may need a mains soft-starting system if you have a really substantial capacitor bank and low-impedance transformer like in a big audio amplifier.

We published a circuit design for limiting inrush currents in the April 2013 issue, the "SoftStarter". PE

MAX'S COOL BEANS

By Max the Magnificent

# WEIRD & WONDERFUL ARDUINO PROJECTS

## Part 9: in which we enjoy a crash course in logic

**I bet I know whare you are thinking** (you weren't expecting that, were you?). Like me, you've likely spent the past month pondering something I wrote in my previous column pertaining to our new all-singing, all-dancing software shift register-based switch debounce solution. I'll remind you about that in a moment, but first...

### Don't blink!

I love Doctor Who. I watched the first episode from behind the safety of the sofa when I was six years old. This aired on Saturday, November 23, 1963, and I remember it as though it were yesterday. I think I've seen every episode since.

I was saddened when the final episode of the classic series aired on December 6, 1989. I was delighted when the first episode of the Doctor Who reboot aired on March 26, 2005. Do you remember the "Blink" episode, which graced our screens on June 9, 2007? This introduced the iconic Weeping Angels, along with the Doctor's admonition: "Whatever you do, don't blink!"

The reason I mention this here is that our latest and greatest switch debounce solution introduces a lag (delay) between a switch being pressed and our taking some action, such as lighting one of our light-emitting diodes (LEDs). If you blink, you'll miss this delay, but what if you don't blink?

And thus, we return to what I said last month. As you may recall, I wrote:

*A delay under 10ms is typically imperceptible to humans. A delay of 10-30ms may be slightly noticeable to very sensitive users, especially in contexts that require fast feedback (eg, gamers or musicians), but it generally still feels instantaneous. A delay of 30-100ms becomes noticeable to the average person, who may describe the system as feeling "laggy" or "slow to respond".*

I went on to write, "... as a rule of thumb, we should aim to keep any button-to-action delays under 50ms". I have faith in this 50ms value because a grizzled old engineer once told me so many moons ago, when I was but a young sprout. And why do **you** have faith in this 50ms value? It's because I've turned into a grizzled old engineer who's telling you so!

I closed that portion of our discussions with, "However, don't take my word for any of this because this is a test you can easily perform for yourself using your Arduino, pushbuttons and LEDs."

Of course, this is when I started to wonder about the veracity of my 50ms pontification, so I thought we could check it out together.

### We're short of nothin' we've got

One of my granddad's go-to sayings was, "We're short of nothin' we've got", which he deployed when he had to make do with something he did have, even if it wasn't ideally suited to the purpose.

Now, we could easily build a standalone breadboard-based test rig using three switches, one LED and any associated resistors. However, we already have three momentary pushbutton switches on our existing shift register breadboard.

If you wish to refresh your memory, you can download a copy of the latest and greatest version of the current state of our board in the file named **CB-sep25-brd-01.pdf** (as usual, all files mentioned in this column are available as part of the September 2025 download package from the PE website at **https://pemag.au/link/ac6z**).

Note that we aren't going to modify the existing linear feedback shift register (LFSR) circuit on our breadboard in any way. We're just going to temporarily change what we do with the switches in software.

For the purposes of this experiment, we can use our left-hand switch to perform the role of "Up" (increase the delay between the switch being pressed or released and the LED changing). Our middle switch can perform the role of "Down" (decrease the delay). And we can use the right-hand switch to trigger actionable events, like activating and deactivating our LED, so we'll call it "Trig."

The Arduino has an onboard orange LED connected to digital input/output (I/O) pin 13, and we can certainly use this. However, it is a little small, so we could optionally add a bigger LED to our breadboard ("I like big LEDs and I cannot lie" – Sir Blinksalot).

This means that the way we are currently visualising the switch portion of our breadboard, including the addition of the optional LED, is as depicted in Fig.1.

### The ironies of life

As you know, we've spent the last two columns developing a super-spiffy switch debounce solution that would make our mothers squeal in delight (if they cared). So, it's a tad ironic that, for this experiment, we need to regress to employ our first debounce solution code from the June 2025 issue.

This is because we need to trigger things from the leading (first) edge in the switch bounce signal. I've copied this program over into the file named **CB-sep25-code-01.txt** to make things easy for you (that's just the sort of fellow I am).

Also, we are going to reuse some

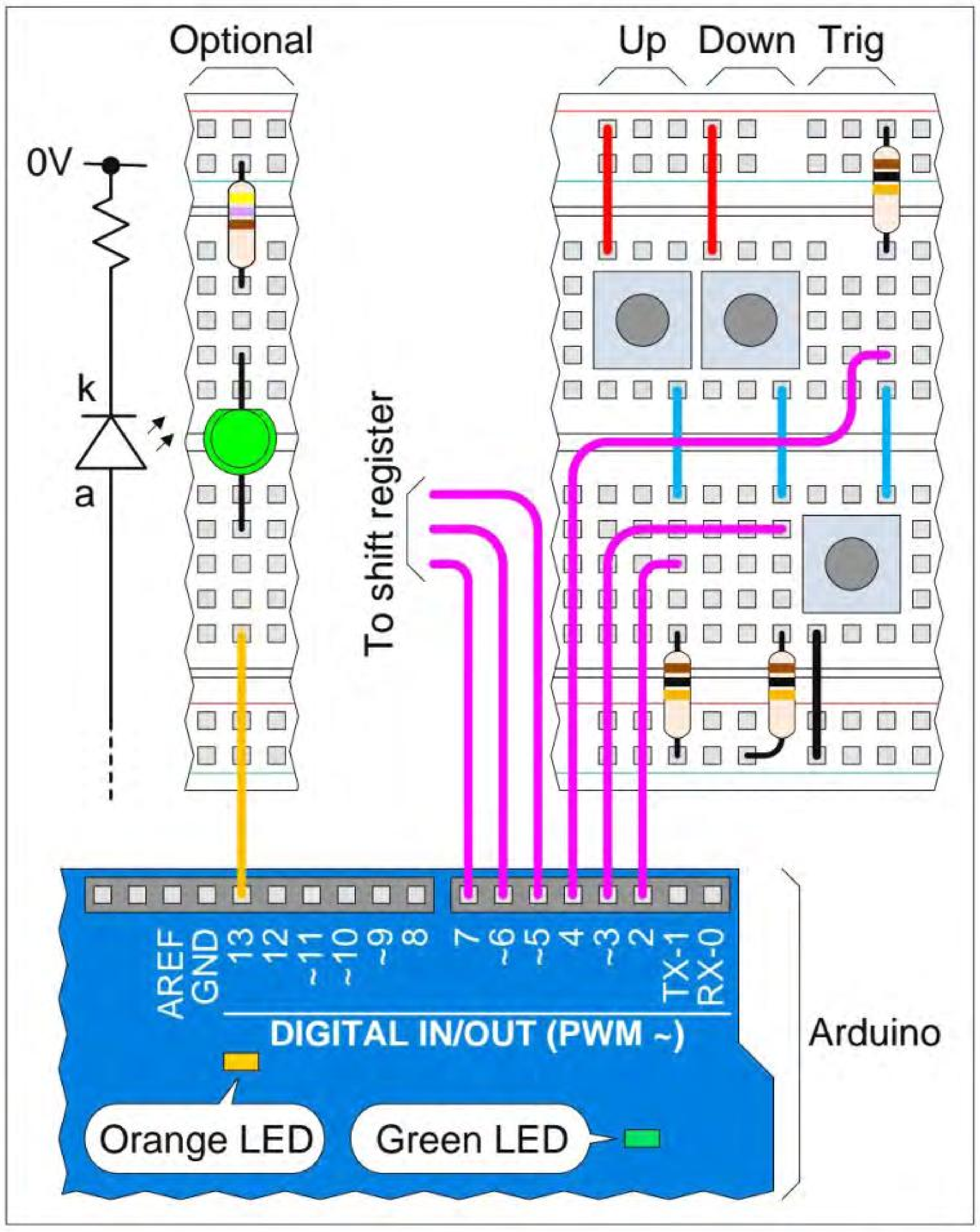


Fig.1: we're adding to and modifying some of our existing breadboard layout, so we can reuse what we already have to save time and effort.

```
1   #define JUST_A_BIT        10
2
3   // SW (switch) signals
4   #define UP_INACTIVE      LOW
5   #define UP_ACTIVE        HIGH
6   #define DOWN_INACTIVE    LOW
7   #define DOWN_ACTIVE      HIGH
8   #define TRIG_INACTIVE    HIGH
9   #define TRIG_ACTIVE      LOW
10
11  // SR (shift register) signals
12  #define RCLK_INACTIVE    LOW
13  #define RCLK_ACTIVE      HIGH
14  #define SRCLK_INACTIVE   LOW
15  #define SRCLK_ACTIVE     HIGH
16  #define SRCLR_INACTIVE   HIGH
17  #define SRCLR_ACTIVE     LOW
18
19  // Display LED stuff
20  #define LED_INACTIVE     LOW
21  #define LED_ACTIVE       HIGH
22
23  // Lag from switch to LED stuff
24  #define INITIAL_LAG        50
25  #define MIN_LAG    JUST_A_BIT
26  #define MAX_LAG          1000
27  #define INC_DEC_LAG         5
```

Listing 4(a): these definitions make the rest of our code easier to understand.

```
29  // Pins from switches
30  int PinUpFromSw   = 2;
31  int PinDownFromSw = 3;
32  int PinTrigFromSw = 4;
33
34  // Pins to shift register
35  int PinSrclrToSr  = 5;
36  int PinSrclkToSr  = 6;
37  int PinRclkToSr   = 7;
38
39  // Orange LED on Arduino board
40  int PinToLed      = 13;
41
42  // Declare global variables
43  int Lag = INITIAL_LAG;
```

Listing 4(b): these are our pin and global variable declarations.

of the definitions and other snippets from the two versions of the shift register code we created last month, which I've copied over as files named **CB-sep25-code-02.txt** and **CB-sep25-code-03.txt** (you're welcome).

I then munged bits and pieces of these three programs together to create our new test program, which you can access in the file named **CB-sep25-code-04.txt**

### Dissecting our test program

It's worth explaining this program in a little detail. As usual, we start with some definitions that will improve the quality of our lives by making our code easier to understand and maintain. These definitions are shown in Listing 4(a). We are using the convention that the listing number (4 in this case) corresponds to the numerical part of the matching code file name ("04" here).

The **JUST_A_BIT** on line 1 is the delay we use to make sure the switch has stopped bouncing after we've done whatever we wanted to do. Lines 4 through 9 are where we define the active and inactive levels associated with the signals we are reading from our three switches. Meanwhile, lines 12 through 17 are where we define the levels associated with the signals we are using to drive the shift register.

"Hang on!", I hear you cry, "I thought we weren't going to use the shift register in this experiment." You're right, we aren't. On the other hand, we don't want to leave the control signals to the shift register floating, because this could potentially cause it to display random, unwanted, and annoying values, so we will make use of these definitions to clear the register and ensure it stays that way.

We use Lines 20 and 21 to define the active and inactive states associated with the orange LED onboard the Arduino, along with the optional LED on the breadboard (I've added one to mine).

Lines 24 through 27 are the definitions associated with the lag (delay) we are going to introduce between the pressing (and releasing) of the switch and the activating (and deactivating) of the LED.

We've set the **INITIAL_LAG** (the one we'll use when the program starts running) to be 50ms. This program will allow us to experiment with different lags that we can set between some minimum and maximum values, which I arbitrarily decided to set to 10 and 1000, respectively.

Based on this, we could have set the **MIN_LAG** to 10ms. However, making it equal to **JUST_A_BIT**, which we've already defined as being 10ms, means that we can never cause the minimum lag to be less than the switch debounce delay.

Last, but not least, we've defined **INC_DEC_LAG** to be 5ms. This is the amount by which we will increment or decrement our lag value when we press the Up or Down buttons, respectively. If you're pedantic, you can make this delay smaller, all the way down to 1ms, but then it will take you a lot of pushbutton presses to get anywhere.

Alternatively, you could make it larger, like 10ms, for example, but then you'll be sacrificing resolution.

Next, as shown in Listing 4(b), we declare the names we are going to use for our pins and assign appropriate pin numbers to them. Also, on line 43, we declare a global variable called **Lag** and assign it a starting value of **INITIAL_LAG**, which we previously defined as being 50ms.

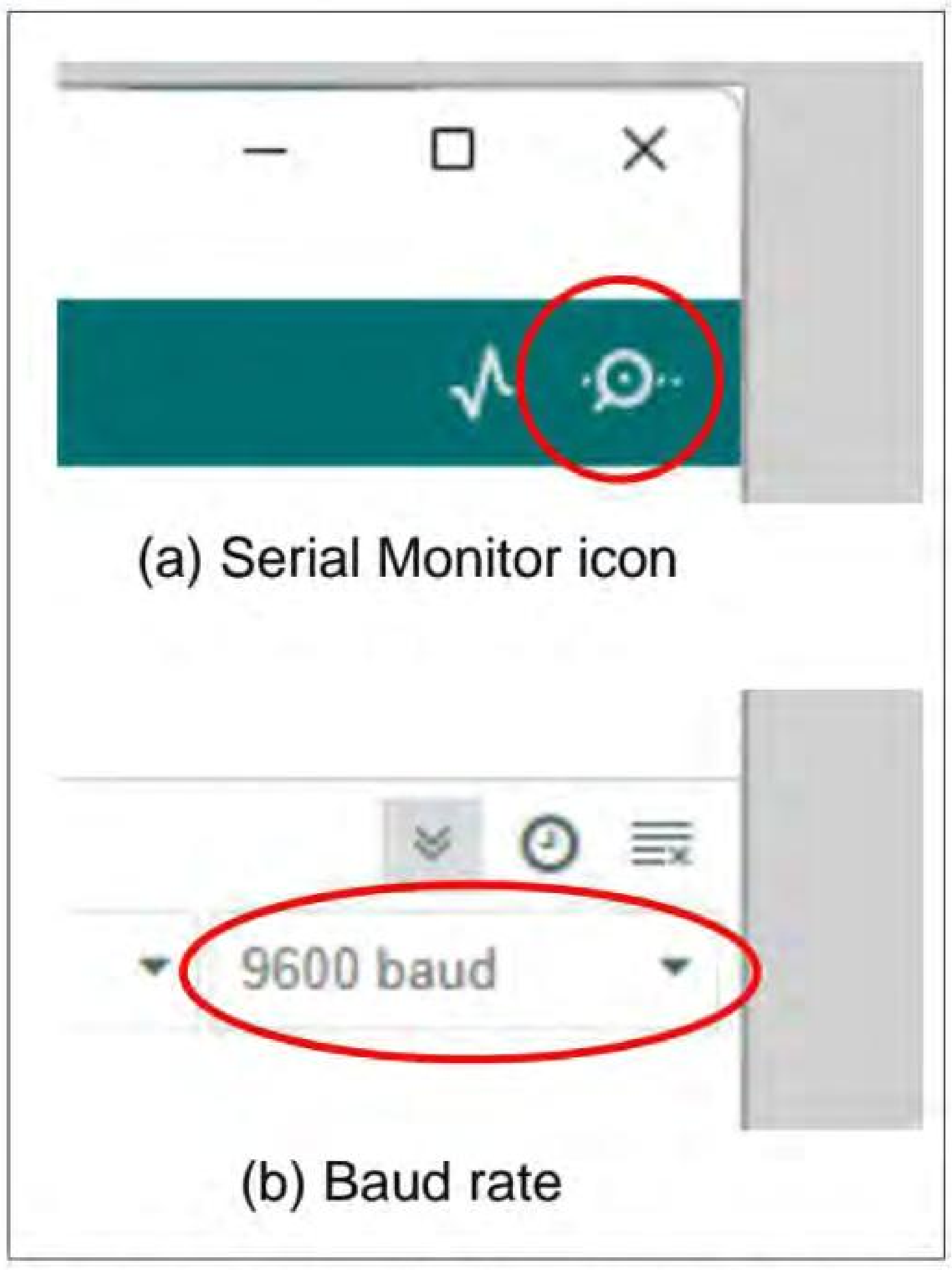
In the case of the **setup()** function (not shown here), the first thing we do is initialise serial communications between our Arduino and host computer using the following statement:

```
Serial.begin(9600);
```

We previously introduced these serial communications in the October

```
85   // Do this over and over again
86   void loop ()
87   {
88     // Increase the Lag
89     if (digitalRead(PinUpFromSw) == UP_ACTIVE)
90     {
91       if (Lag < MAX_LAG) Lag += INC_DEC_LAG;
92       DisplayLag();
93       delay(JUST_A_BIT);
94
95       while (digitalRead(PinUpFromSw) == UP_ACTIVE)
96       {
97       }
98       delay(JUST_A_BIT);
99     }
100
101    // Decrease the Lag
102    if (digitalRead(PinDownFromSw) == DOWN_ACTIVE)
103    {
104      if (Lag > MIN_LAG) Lag -= INC_DEC_LAG;
105      DisplayLag();
106      delay(JUST_A_BIT);
107
108      while (digitalRead(PinDownFromSw) == DOWN_ACTIVE)
109      {
110      }
111      delay(JUST_A_BIT);
112    }
113
114    // Respond to trigger switch
115    if (digitalRead(PinTrigFromSw) == TRIG_ACTIVE)
116    {
117      delay(Lag);
118      digitalWrite(PinToLed, LED_ACTIVE);
119
120      while (digitalRead(PinTrigFromSw) == TRIG_ACTIVE)
121      {
122      }
123      delay(Lag);
124      digitalWrite(PinToLed, LED_INACTIVE);
125    }
126  }
```

Listing 4(c): this main loop is where all the magic takes place.


(a) Serial Monitor icon

9600 baud

(b) Baud rate

Fig.2: the Arduino Serial Monitor window.

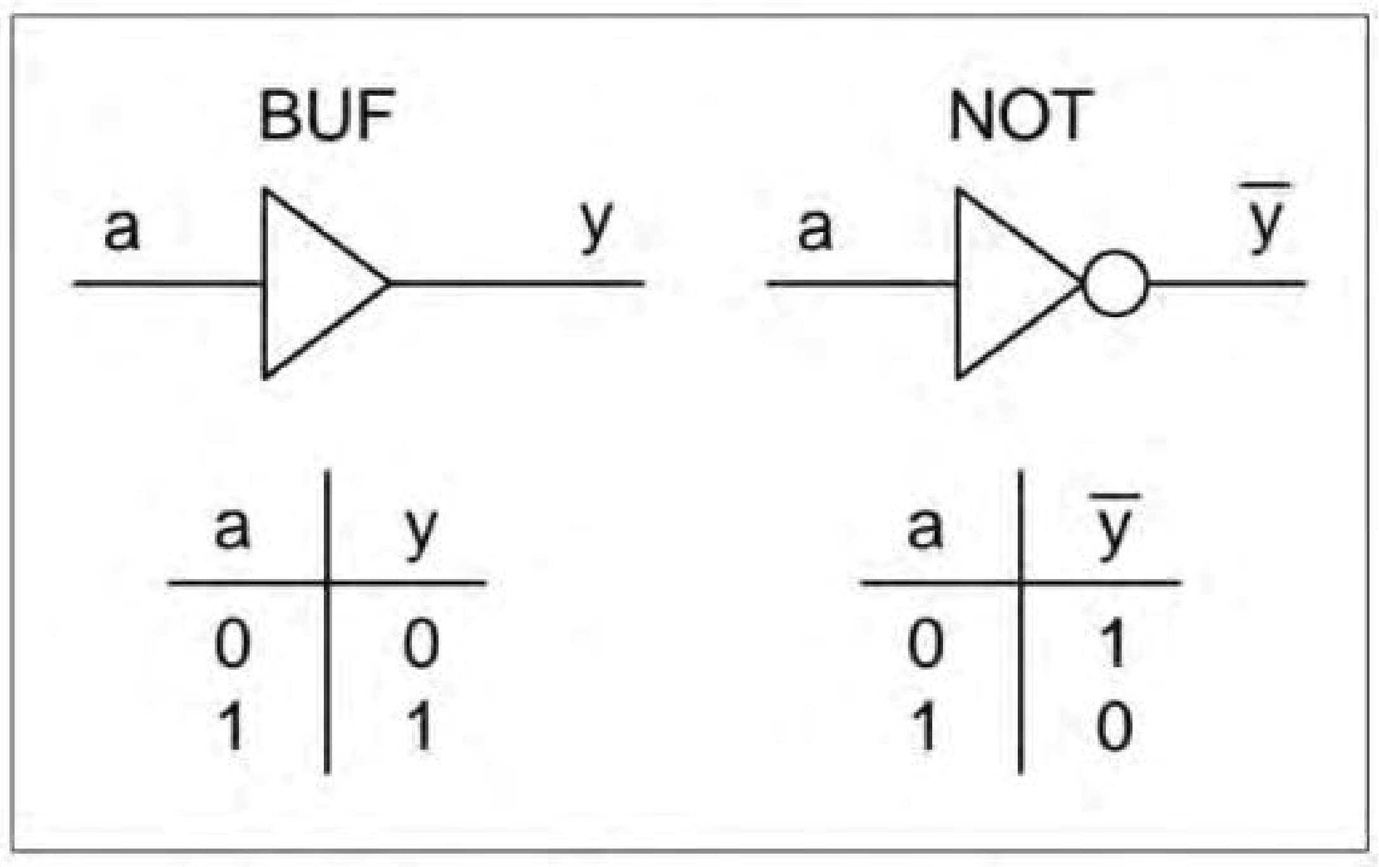


Fig.3: BUF and NOT (inverter) gates.

2023 issue of our *Arduino Bootcamp* series. As a brief reminder, the serial port (usually over USB on most Arduino boards) can be used to send data to and from the Arduino and our host computer, often for debugging or communication with other devices.

The 9600 value is the default baud rate, the speed at which data is transmitted and received. For our purposes, we can consider the baud rate as being the number of bits being transmitted per second.

Next, we specify the modes (**INPUT** or **OUTPUT**) associated with our pins, set the outputs to their inactive states, and then use the appropriate signals to clear the shift register on our breadboard so we won't get distracted by errant segments being activated on our 8-segment display.

Finally, we call a little function called **DisplayLag()** (not shown here) that you'll find at the end of the program. This function uses serial commands to display the current value of **Lag** on the screen of our host computer.

Now, let's turn our attention to the interesting stuff, which we find in the **loop()** function, illustrated in Listing 4(c).

Starting on line 89, we check to see if our Up switch has been pressed. If so, we add **INC_DEC_LAG** to the current **Lag** value, call our **DisplayLag()** function to display the new value on our host computer, then we wait for the switch to be released.

Similarly, starting on line 102, we check to see if our Down switch has been pressed. If so, we subtract **INC_DEC_LAG** from the current **Lag** value, display the new value on our host computer, then we wait for the switch to be released.

Starting on line 115, we check to see if the **Trig** has been pressed. If so, we pause for the **Lag** time, then we activate the LED. Then, starting on line 120, we wait for the **Trig** switch to be released. When this occurs, we pause for the **Lag** time, and then we deactivate the LED.

### Running our test program

While we are writing and verifying our programs, the top half of the screen in the Arduino's integrated development environment (IDE) shows our source code. Meanwhile, the bottom half of the screen is the console area that typically shows errors and warning messages when we compile our code, along with upload progress and results, like "Done Uploading".

Plug the USB cable into the Arduino to power everything up and then upload our test program. Next, click the Serial Monitor icon located in the top right-hand corner of the IDE, as seen in Fig.2(a).

Also look at the field in the upper right-hand corner of the console area and make sure that the host computer's baud rate is set to 9600 (the same value we specified in our program), as in Fig.2(b). Both devices (Arduino and computer or serial monitor) must use the same baud rate to communicate properly.

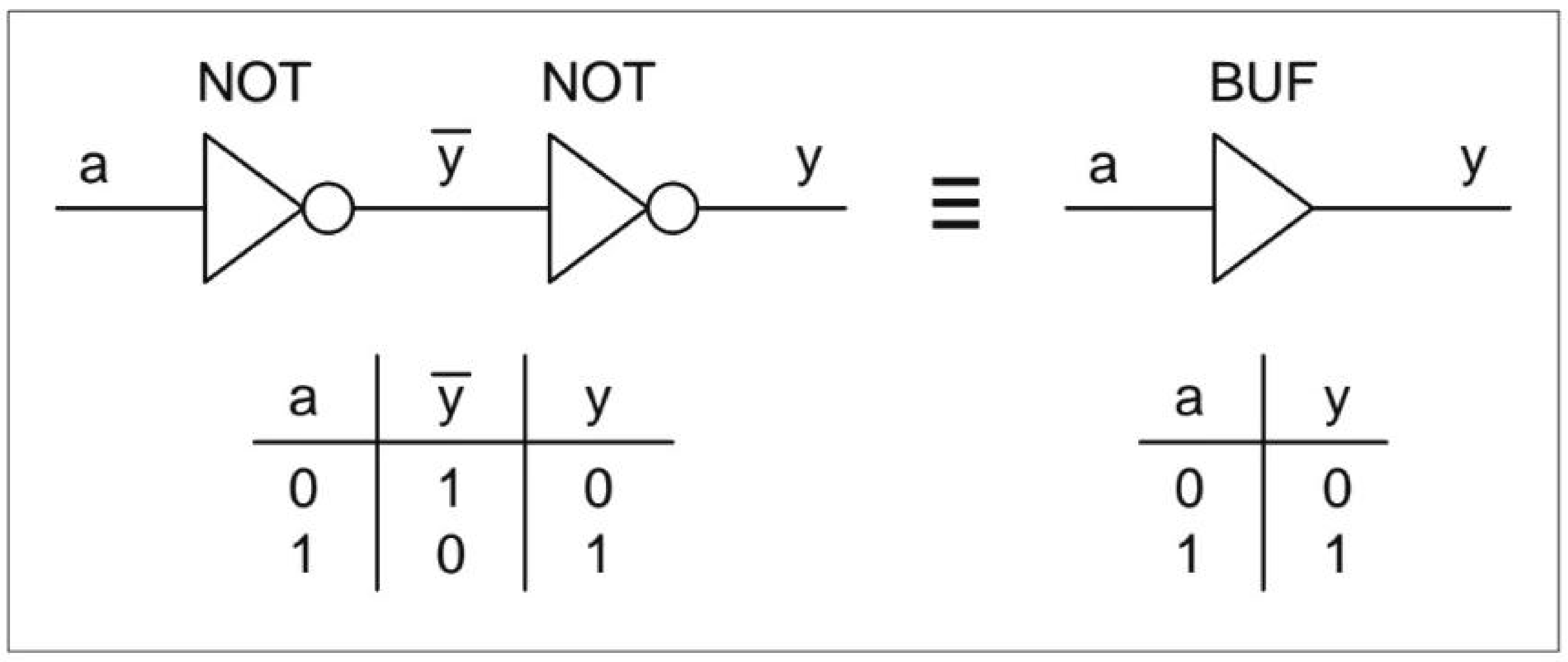
You should see something like the following displayed in your console area:

```
Lag =  50ms
```

Press the Up button a few times and observe the lag value increase in 5ms increments:

```
Lag =  50ms
Lag =  55ms
Lag =  60ms
Lag =  65ms
```

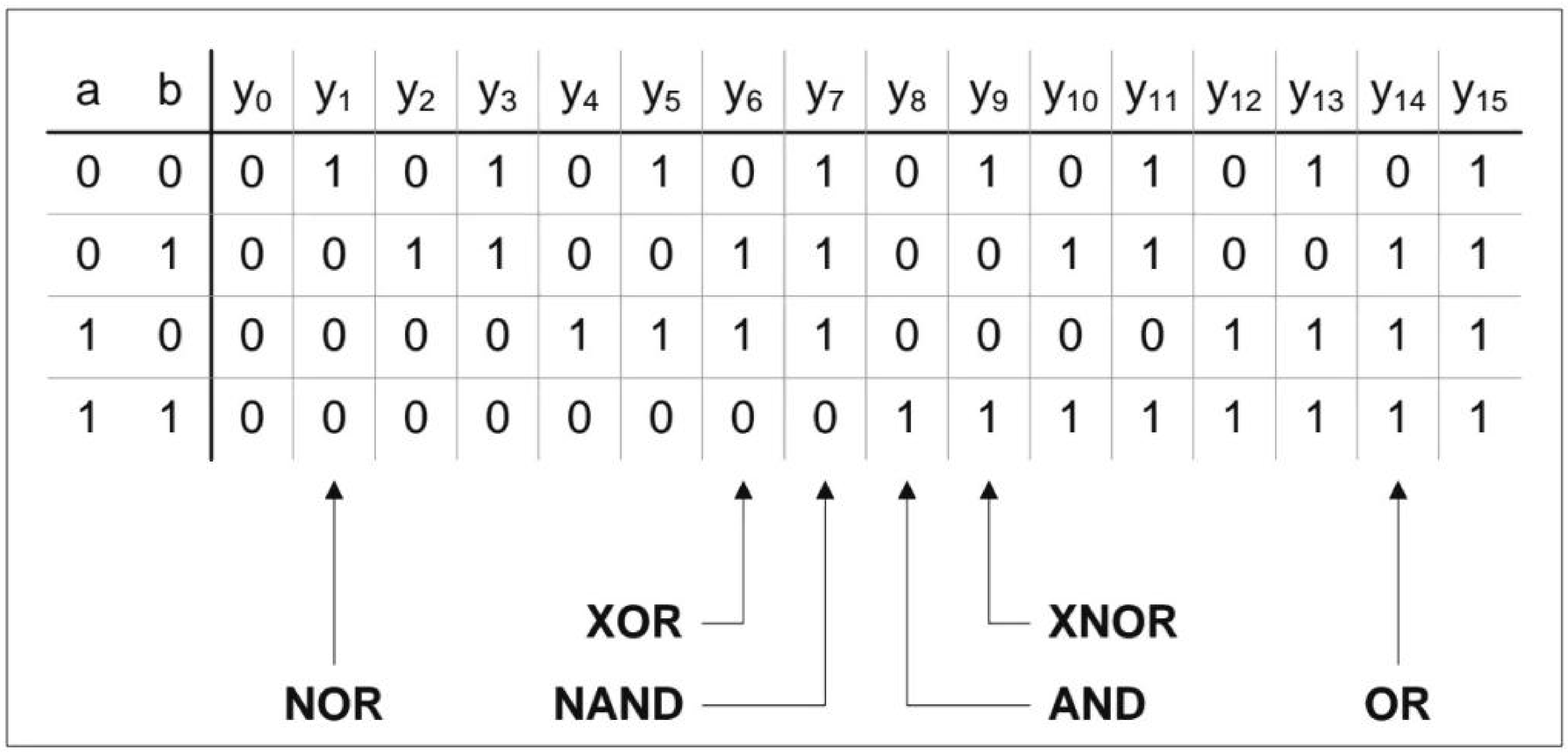
Next, press the Down button the same number of times to return the lag value to its original 50ms value.



| a | $\overline{y}$ | y |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 0 | 1 |

| a | y |
|---|---|
| 0 | 0 |
| 1 | 1 |

Fig.4: two NOT gates connected in series are equivalent to a single BUF gate.



| a | b | $y_0$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ | $y_6$ | $y_7$ | $y_8$ | $y_9$ | $y_{10}$ | $y_{11}$ | $y_{12}$ | $y_{13}$ | $y_{14}$ | $y_{15}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

XOR
XNOR
NOR
NAND
AND
OR

Fig.5: a logic gate with two single-bit inputs has 16 potential output permutations.

Now for the fun part. Press and hold the Trig button and observe the orange LED light up on the Arduino board (and the optional big LED on your breadboard, if you added one). Then release the Trig button and observe the LED(s) turn off.

Did you perceive any delay between pressing (or releasing) the button and change in the LED? Before performing this experiment, I honestly believed that I'd be able to perceive a lag of 50ms, but such was not the case. Maybe I could have done so when I was a younger man, but the years have not been kind.

I'm embarrassed to tell you how high I had to make the lag value before I could spot a delay between pushing the button and observing a response, so I won't. Now I'm going to try this on people of varying ages, including my wife (Gigi the Gorgeous) and the kids who live next door. I'd be interested to hear about your own experiences here; I will report back further next month.

### Homework assignment #1

Since we're here, why don't we create a complementary "Reaction Tester" experiment using the orange LED (and optional LED) and one or more of our pushbutton switches?

The idea is for the Arduino to wait some random amount of time before illuminating the LED(s). As soon as we see the LED light up, we press the button, and the Arduino displays the time it took for us to respond on our host computer. How about if you go first, and then I'll present my solution to this assignment in next month's column?

### That's logical

Now, hold on to your hat, because this next part is going to be tremendously exciting (have I ever lied to you before?). In the July 2025 issue, I posed a couple of logic problems pertaining to the implementation of our LFSR. I'll prompt you regarding those posers presently. First, however, we need to delve a little deeper into the subject of primitive logic gates.

Let's start with the low-hanging fruit, which are gates with only one input and one output: BUF and NOT (aka. INV), as illustrated in Fig.3. Observe the 'bobble' (or 'bubble') on the output of the NOT gate. This indicates an inverting function.

Also observe that we are using horizontal lines over output signal names (eg, $\overline{y}$) to indicate their inverted attribute. This is called "overbar notation" or "bar notation". It's common to say "Y-bar" in verbal communication, or when thinking about it in your head (which, in my experience, is the best place to think about things).

An alternative would be to employ a purely textual terminology, like **yb**, where the 'b' stands for "bar".

Remember that the 0s and 1s in these truth tables equate to voltages in the real world: 0V and +5V, respectively, in the case of the integrated circuits (ICs) we are working with.

When I was a young sprog starting out in digital electronics, I really didn't have a clue (not that things are much better now). I remember how the truth table for the BUF gate made sense, but I found it hard to wrap my brain around the truth table for the NOT. How could a 0 (0V) on the input result in a 1 (+5V) on the output, for example?

I didn't realise that these gates also have 0V and +5V power rails that are separate from the inputs and outputs, but that they are not shown in our symbols to keep things simple.

### It's not wrong

I'm sure you've heard the expression, "two wrongs don't make a right" (but three lefts do). Well, that's as may be, but it's certainly true that two NOTs connected in series (one after the other) are functionally equivalent to a BUF (Fig.4).

I don't know about you, but I tend to think of non-inverting gates like BUF as being simpler than inverting functions like NOT. This may be true conceptually, but it's not the case in actuality.

With the CMOS (complementary metal-oxide semiconductor) technology that is used to create most of today's digital ICs, a NOT gate requires two transistors. A BUF gate is essentially formed from two NOT functions in series (as in Fig.4), so it requires four transistors. Furthermore, a NOT gate has only one stage of delay (so it's faster), while a BUF gate has two stages of delay (so it's slower).

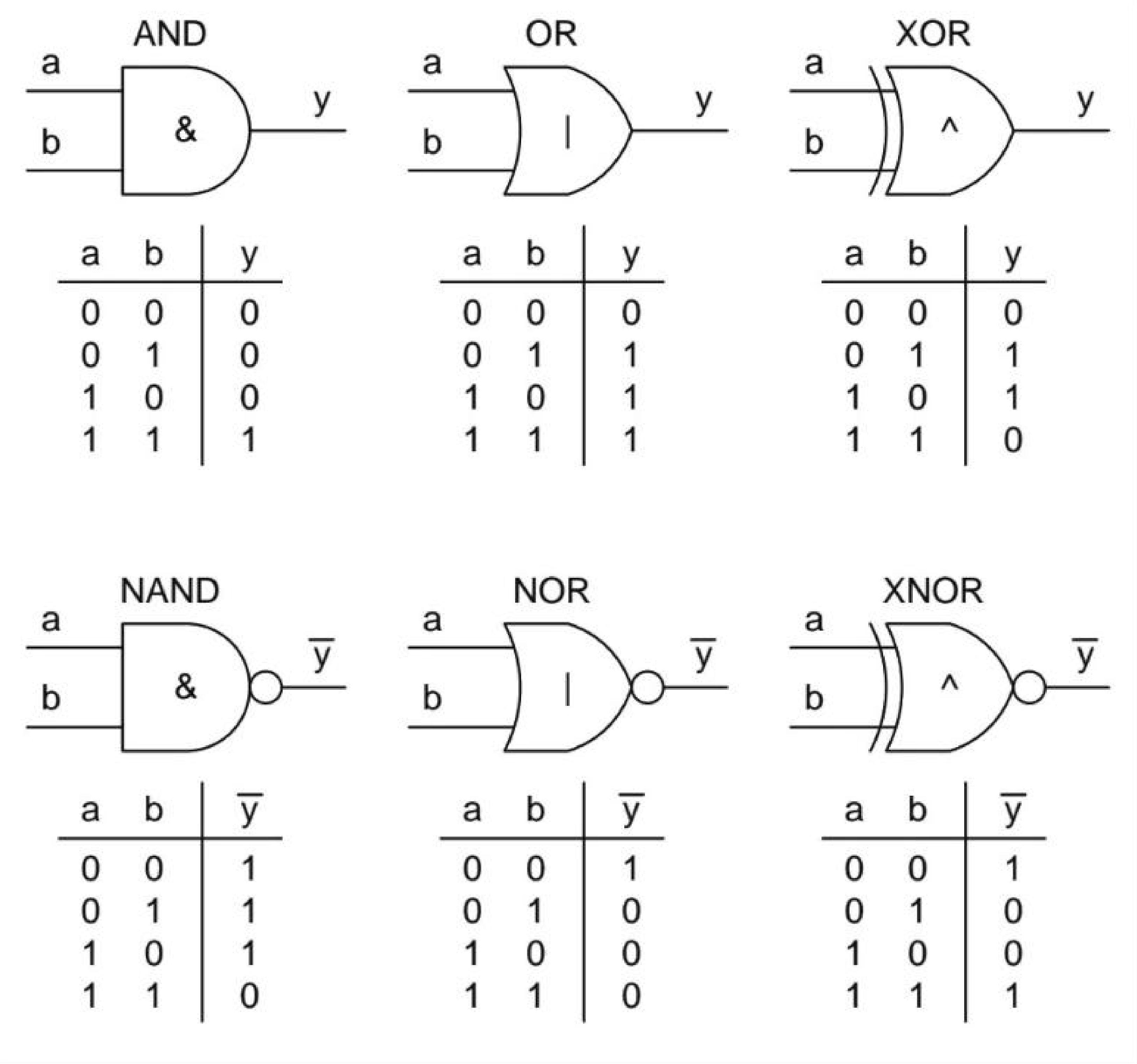
In fact, it's generally fair to say that inverting functions are simpler and faster than their non-inverting counterparts in any implementation technology (electronic, hydraulic, mechanical, pneumatic…).

If you feel an overwhelming desire to discover how these gates are built from transistors, this is one of the topics I talk about in my book, *Bebop to the Boolean Boogie* (https://pemag.au/link/abzi).

### A preponderance of permutations

Now let's 'up our game' by moving to gates with two inputs and one output. Let's call the inputs **a** and **b**. These inputs have $2^2 = 4$ possible combinations of 0s and 1s: 00, 01, 10, and 11. This means there are $2^4 = 16$ possible permutations; that is, 16 different ways to assign a 0 or 1 output to each of the four input combinations. Let's call these permutations $y_0$ through $y_{15}$, as shown in Fig.5.

Depending on what we are trying to achieve at the time, we could end up employing any of these permutations in a design. In practice, however, we use some more often than others, so



AND

| a | b | y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

OR

| a | b | y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

XOR

| a | b | y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

NAND

| a | b | $\overline{y}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

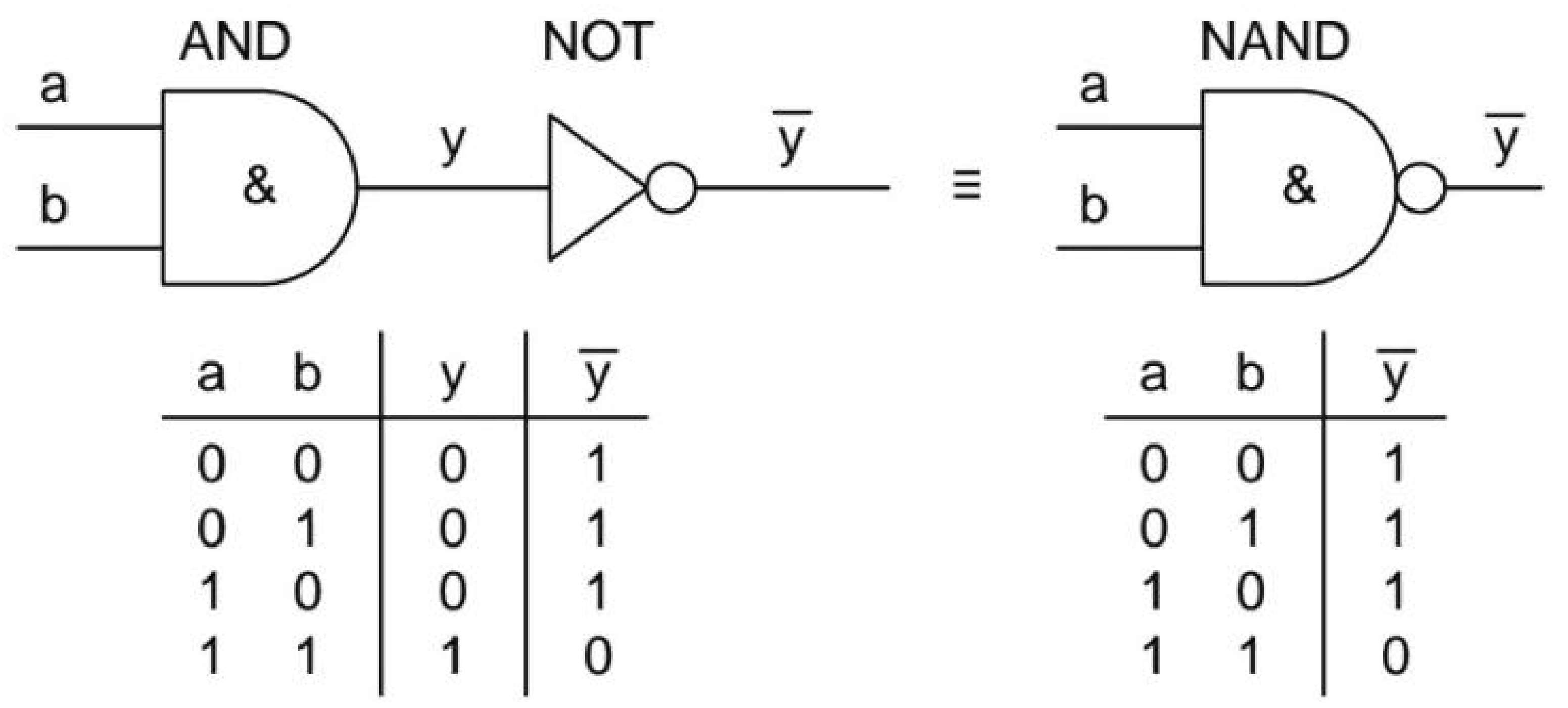
NOR

| a | b | $\overline{y}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

XNOR

| a | b | $\overline{y}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

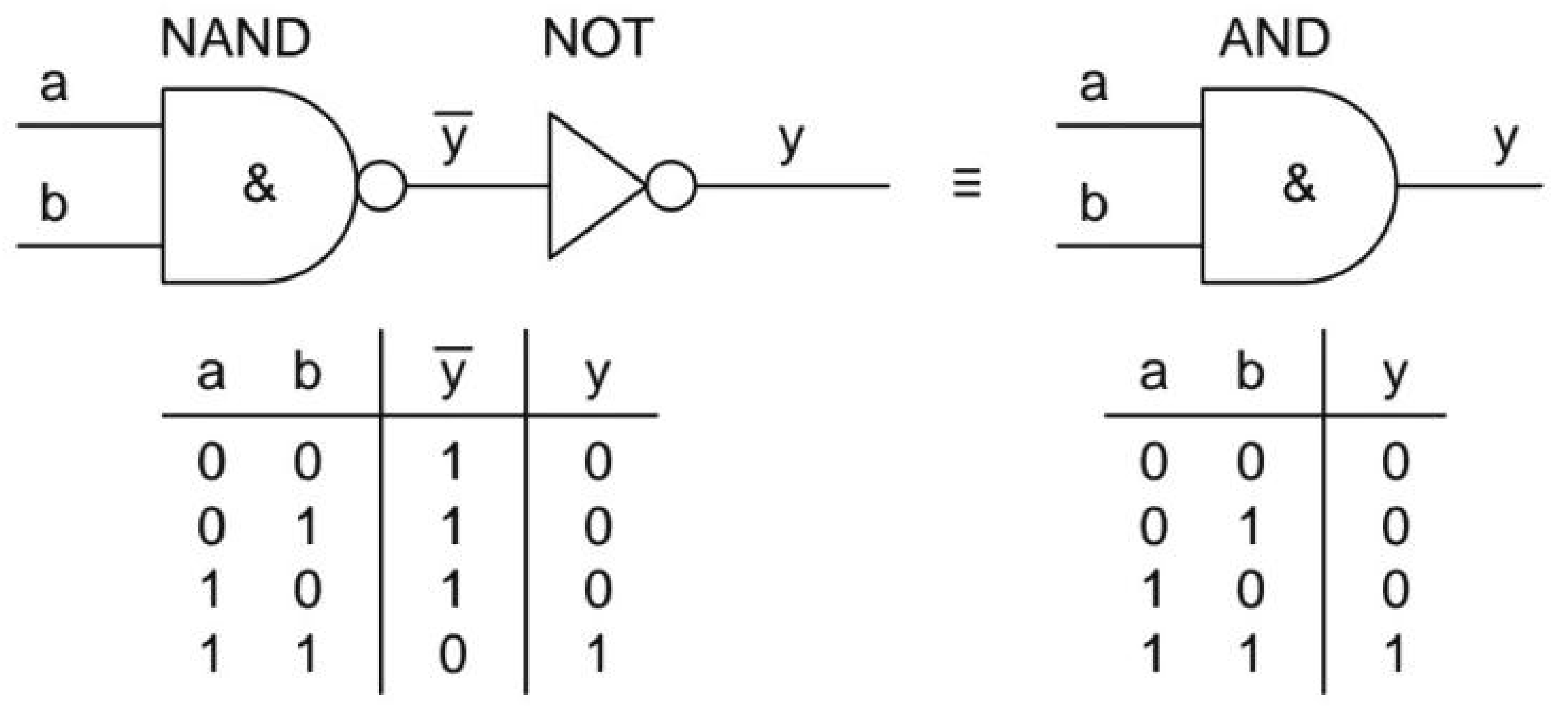
Fig.6: symbols & truth tables for the named 2-input logic functions.



| a | b | y | $\overline{y}$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

| a | b | $\overline{y}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

(a) Making a NAND by inverting an AND



| a | b | $\overline{y}$ | y |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

| a | b | y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

(b) Making an AND by inverting a NAND

Fig.7: inverting AND and NAND logic gates.

we give these special names: AND, OR, XOR, NAND, NOR and XNOR. The symbols and truth tables for these named 2-input logic functions are shown in Fig.6.

## Bending over backwards

I'm afraid the next bit involves some mental gymnastics (I'll bend over backwards to be flexible). We take NAND and NOR to mean "not AND" and "not OR," respectively, and this is the way we tend to think about things.

It's certainly true that we can make a NAND by inverting the output of an AND, as illustrated in Fig.7(a). Contrariwise, we can make an AND by inverting the output of a NAND, as illustrated in Fig.7(b). We can do the same with OR and NOR gates, but I'll leave that as 'an exercise for the reader'.

One way to visualise what's happening in Fig.7(a) is that we 'push' the NOT gate symbol back into the AND gate symbol until only its bobble remains visible. Similarly, one way to envisage what's happening in Fig.7(b) is that we 'push' the NOT gate back into the NAND, and the two bobbles cancel each other out.

The funny thing is that Fig.7(a) is the way we usually think about things, but Fig.7(b) is the way things are in the real world. As with NOT and BUF gates, it's easier to create an inverting function than its non-inverting counterpart.

In the case of a NAND gate implemented in CMOS, each input requires two transistors, so a two-input NAND employs four transistors. Since a NOT gate requires two transistors, a 2-input AND gate, which is formed from a NAND and a NOT, consumes six transistors. Furthermore, a NAND gate has only one stage of delay (so it's faster), while an AND gate has two stages of delay.

This may be a good time to note that 'slower' is relative, because all these electronic gates are screamingly fast. With respect to what we are doing in our experiments, any differences in delay between an AND and a NAND are infinitesimal and imperceptible. This really becomes of interest only to people designing far more complex chips or circuits.

## Tricky XORs

One question many people ask is why we say XNOR ("exclusive NOR") instead of NXOR ("Not XOR"). In fact, some people do use these terms interchangeably, but there is a subtle distinction.

The problem is that saying NXOR explicitly emphasises the NOT operation on an XOR. However, we don't actually need to apply a NOT gate to an XOR gate to generate a NXOR gate (although we could if we wanted to) because an XNOR is a defined gate in its own right. To put this another way, XOR and XNOR gates use the same number of transistors, and both have only one stage of delay.

There are multiple implications that stem from this. Let's start with the fact that it's possible for AND and OR (and NAND and NOR) gates to have more than two inputs. Remember that each input to a NAND gate requires two transistors, so a 4-input NAND gate would employ

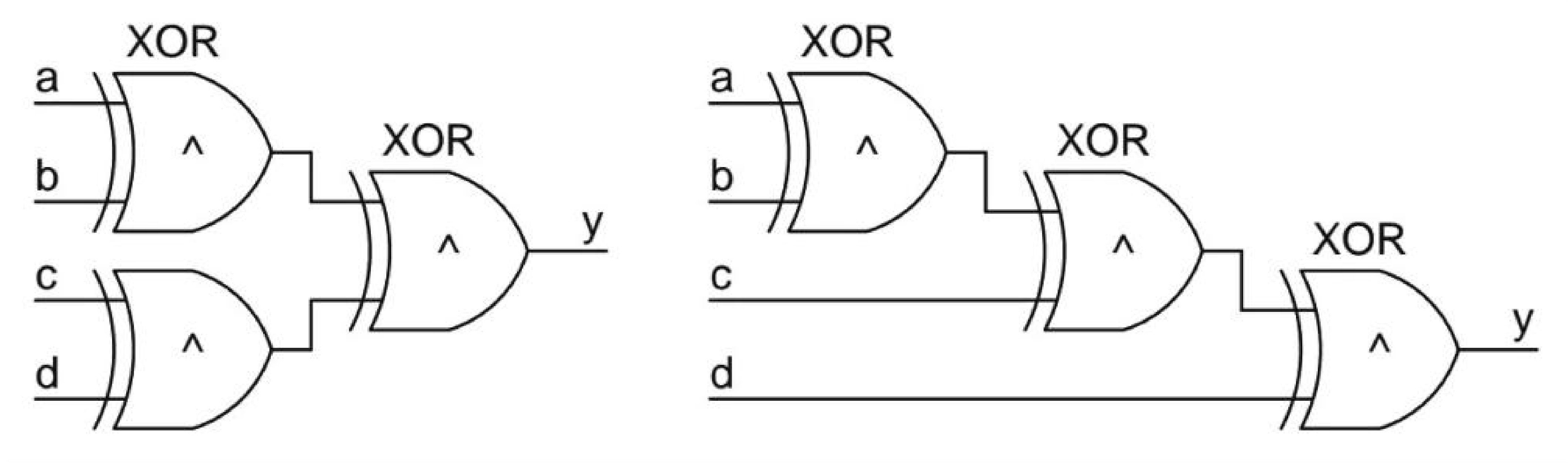


Fig.8: two ways to implement a 4-input XOR function.

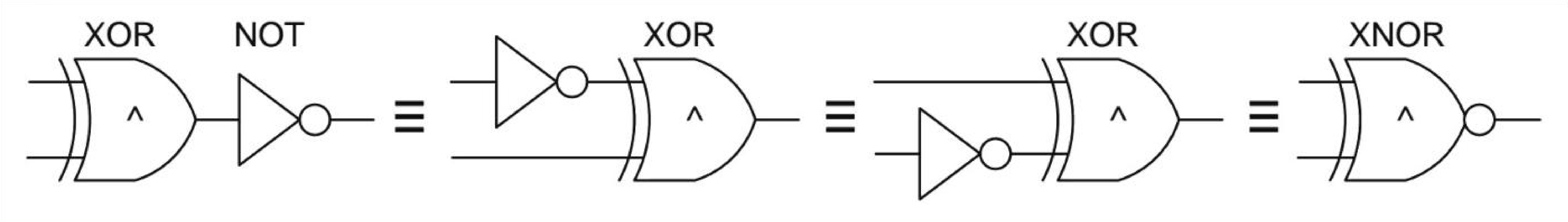


Fig.9: 3 ways to make an XNOR using XOR & NOT.

eight transistors, while its 4-input AND counterpart would consume ten transistors.

By comparison, physical XOR and XNOR gates can only ever have two inputs. So, what do we do if we require an XOR function with more inputs? Well, we just need to use multiple 2-input XOR gates. For example, two ways to implement a 4-input XOR function using three 2-input XOR gates are illustrated in Fig.8.

As we previously discussed, we can generate a NAND by inverting the output of an AND, but not by inverting either of its inputs. Similarly, we can generate a NOR by inverting the output of an OR, but not by inverting either of its inputs.

By comparison, we can generate an XNOR from an XOR by inverting its output or either of its inputs, as per Fig.9. Similarly, we can generate an XOR from an XNOR by inverting its output or either of its inputs.

### Parity checking

One of the simplest forms of error detection used in digital communication and storage systems is known as 'parity checking'. This involves adding an extra bit, called a parity bit, to a set of other bits to ensure the integrity of that data during transmission or storage.

There are two types of parity: even parity and odd parity. Even parity is where the parity bit is set so that the total number of 1s in the entire binary word (including the parity bit) is even. In the case of odd parity, the parity bit is set so that the total number of 1s is odd.

Consider a communications channel. The transmitter will generate a parity bit for each word it transmits. This bit will be transmitted along with the word of data. The receiver will also generate a parity bit for each word it receives, and it will compare its parity bit to the one from the transmitter.

If the data is inadvertently altered (eg, by electromagnetic noise), causing a single data bit or the parity bit to 'flip', the parity check at the receiver will fail, indicating an error.

An 8-bit word is called a byte. A 4-bit word is called a nybble. The idea that two nybbles make a byte shows that electronic and computer engineers do have a sense of humour, albeit a rudimentary one.

The even and odd parity values associated with a 4-bit data nybble are shown in Fig.10. How would we go about generating these values? Well, let's start by looking at the top four values in the even parity column: 0, 1, 1, 0. Hmm, this is the same sequence as the output from a 2-input XOR. How about the next four values in the even parity column: 1, 0, 0, 1? Gosh, that's the same sequence as for an XNOR.

It doesn't take us long to realise that we can generate an even parity bit that we can call p by simply feeding the four data bits into either of the 4-input XOR functions we described in Fig.8.

Furthermore, as per our discussions pertaining to Fig.9, we could generate an odd parity bit, that we can call $\overline{p}$, by inverting (adding a NOT gate to) the output or to any of the inputs to our 4-input XOR function. Actually, we can generate the $\overline{p}$ signal by inverting multiple inputs and outputs to our 4-input XOR function, just so long as the total number of inversions is odd.

### Little logic gates

There are hundreds of different 7400 device types (you can see a list at https://w.wiki/EY6R). In the case of devices containing only primitive logic gates, we have made use of three types in our experiments thus far (Fig.11).

|  |  | Parity |  |
|---|---|---|---|
| Data | # 1s | Even | Odd |
| 0 0 0 0 | 0 | 0 | 1 |
| 0 0 0 1 | 1 | 1 | 0 |
| 0 0 1 0 | 1 | 1 | 0 |
| 0 0 1 1 | 2 | 0 | 1 |
| 0 1 0 0 | 1 | 1 | 0 |
| 0 1 0 1 | 2 | 0 | 1 |
| 0 1 1 0 | 2 | 0 | 1 |
| 0 1 1 1 | 3 | 1 | 0 |
| 1 0 0 0 | 1 | 1 | 0 |
| 1 0 0 1 | 2 | 0 | 1 |
| 1 0 1 0 | 2 | 0 | 1 |
| 1 0 1 1 | 3 | 1 | 0 |
| 1 1 0 0 | 2 | 0 | 1 |
| 1 1 0 1 | 3 | 1 | 0 |
| 1 1 1 0 | 3 | 1 | 0 |
| 1 1 1 1 | 4 | 0 | 1 |

Fig.10: parity values for four-bit data.

As part of our *Arduino Bootcamp* columns from January 2023 through December 2024, we introduced the 74xx04, which contains six NOT gates, and the 74xx08, which contains four two-input AND gates. In our current *Weird and Wonderful* series, we reused the 74xx04 and added the 74xx86, which contains four two-input XOR gates.

For the sake of reference, other devices of this ilk include the 74xx00 (quad two-input NAND), 74xx02 (quad two-input NOR) and 74xx32 (quad two-input OR).

Remember that the "xx" in these part numbers represents the underlying logic technology used inside the chip. Two commonly used technologies are "LS" (low-power Schottky) and "HC" (high-speed CMOS).

When building circuits, it's generally best to stick to a single logic family. Personally, I prefer the HC variants because they're more tolerant

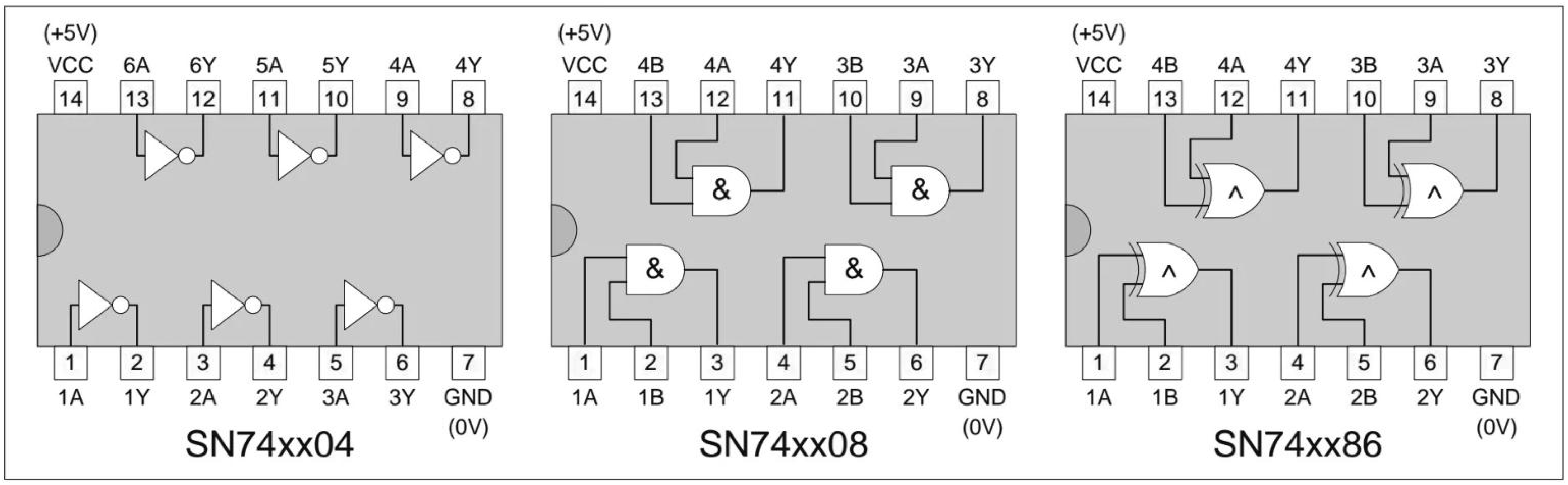


Fig.11: three of the 'primitive' logic chips we've used before. They each contain multiple gates (four or six) for our convenience.

of supply voltage variations, both in terms of ripple and slow drift. That said, it's perfectly fine to use HC outputs to drive LS inputs.

However, using LS outputs to drive HC inputs can be a problem, since the LS output high voltage may not meet the HC input threshold for a logic high. Having said this, there shouldn't be any problem mixing LS and HC devices for what we're doing with our specific breadboard setup, but it would be frowned upon by professional engineers (we'll discuss this in more detail next month).

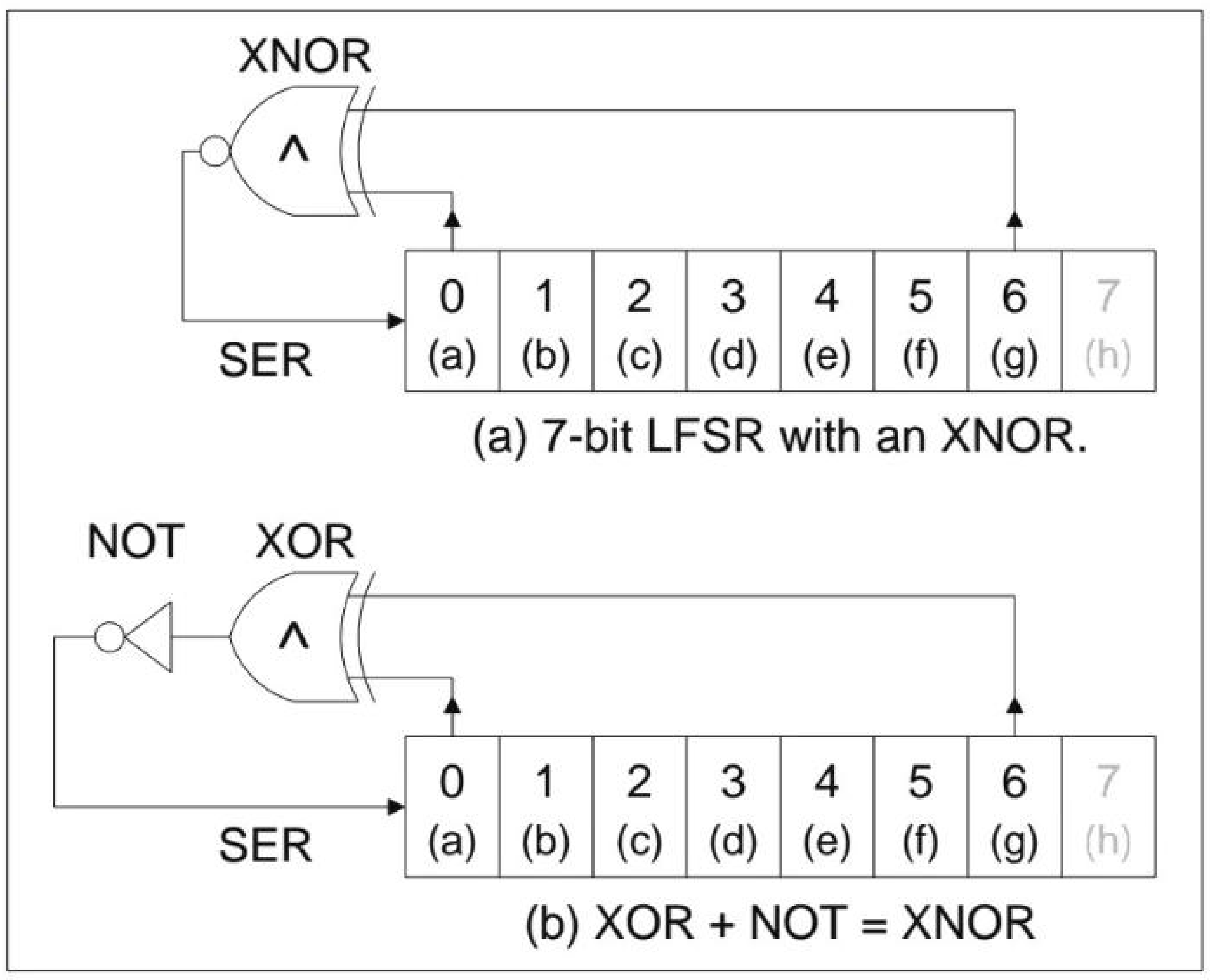
### Our XNOR-based 7-bit LFSR

This brings us back to the homework assignments I posed in the July 2025 issue. Before we address these, let's first remind ourselves that we started by introducing the concept of LFSRs. Next, we decided to create a 7-bit LFSR using a single two-input XNOR gate, as illustrated in Fig.12(a).

The reason for using XNOR rather than an XOR is that an XOR-based LFSR cannot cycle through the all-0s state (it would get locked into the all-0s value). This is unfortunate because we start by clearing our 74HC595 shift register to all 0s. Happily, an XNOR-based LFSR can cycle through the all-0s state (it's the all-1s state it doesn't like).

Sad to relate, we didn't have a two-input XNOR gate in our kit of parts. What we did have was a two-input XOR in the form of our 74xx86 device. So, we inverted the output of this XOR using one of the NOT gates in our 74xx04 to realise an XNOR, as shown in Fig.12(b).

Fig.12: an XNOR-based 7-bit LFSR.



(a) 7-bit LFSR with an XNOR.

(b) XOR + NOT = XNOR

### What? No NOTs?

For the first part of this assignment, I wrote, "... suppose we had only our 74xx86 chip containing its four 2-input XOR gates (so, no 74xx04 with its six NOT gates, and no 74xx08 with its four 2-input AND gates). Can you think of a way to implement the desired XNOR function?"

This one is easy peasy lemon squeezy. Observe the truth table for the 2-input XOR gate in Fig.6. If we connect one of these inputs to logic 1, the gate will invert the other input, which means it's acting as a NOT, as illustrated in Fig.13(c).

### What? No XORs?

Things get a little trickier when it comes to the second part of this assignment, where I wrote, "... suppose we had only our 74xx04 with its six NOT gates and our 74xx08 with its four 2-input AND gates (so, no 74xx86 with its four 2-input XOR gates). Can you think of a way to implement the desired XNOR function?"

There are various ways in which we can achieve this. For example, let's start by implementing an XOR using four 2-input NAND gates as shown in Fig.14(b).

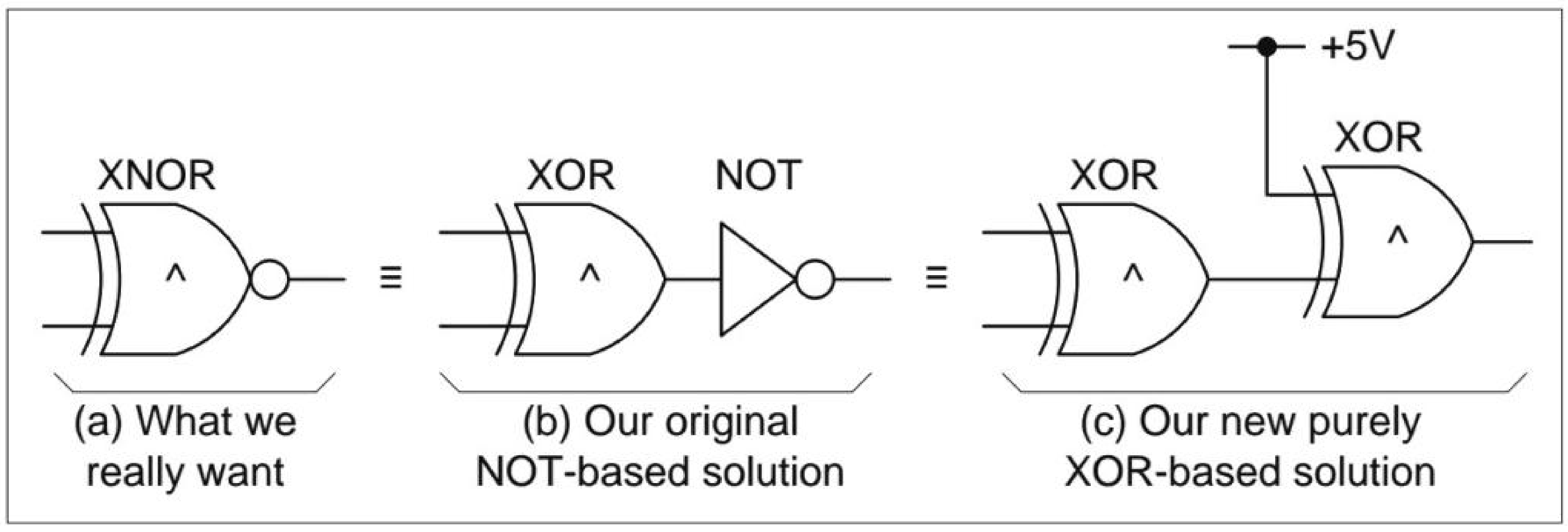
Your first thought might be that this is all very nice, but we don't have any NAND gates. My response would be that we can implement a NAND using an AND and a NOT, as illustrated in Fig.7(a). We can think of this as AND → NOT.

Of course, your next thought will probably be that it's all very well our creating an XOR, but what we really need is an XNOR, as illustrated in Fig.12(a). This is where things get interesting. We still have two NOT gates left, so we could use one to invert the output **y** from our final NAND gate to give the $\overline{y}$ signal we really desire. We can think of this as NAND → NOT.

On the other hand, remember that we are forming our NANDs using ANDs followed by NOTs. This means each NAND → NOT is actually implemented as AND → NOT → NOT.

From our previous discussions, we know that the two NOTs cancel each other out (logically speaking), so all we need to do to achieve our XNOR is to not include the NOT on the gate driving the $\overline{y}$ output (this is great because not doing something is one of the things I do best). The result is illustrated in Fig.15.

I know all of this can be a little brain-boggling at first, but logic design engineers love this sort of thing. Sad to relate, most of these manipulations are performed automatically by design tools these days. On the other hand, it's always useful to know how to do this ourselves, especially when



(a) What we really want

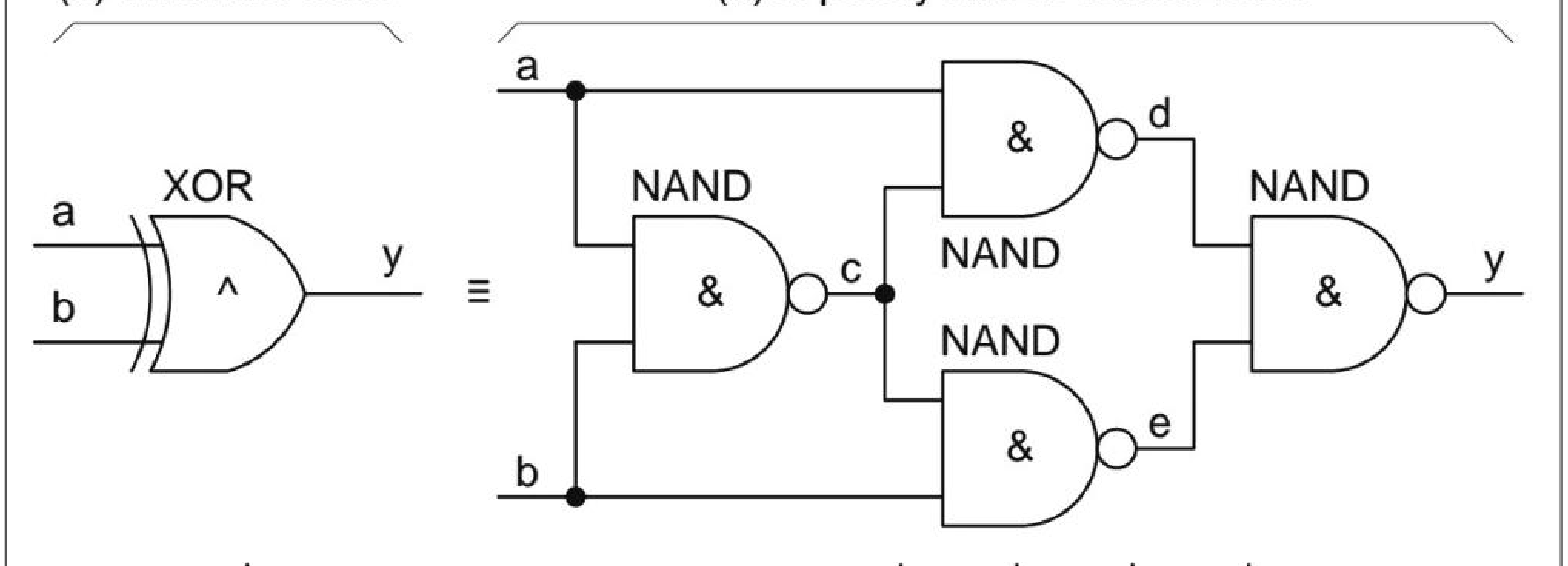
(b) Our original NOT-based solution

(c) Our new purely XOR-based solution

Fig.13: implementing the XNOR function with two XOR gates.

(a) What we want

(b) A purely NAND-based XOR



| a | b | y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| a | b | c | d | e | y |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 |

Fig.14: Implementing the XOR function with four NAND gates.

we are working at the 7400-series logic-chip level.

### Homework assignment #2

I recently went to see the latest Mission Impossible film with a group of friends. This prompts me to say, "Your mission, should you choose to accept it, is to implement the 'No NOTs' and 'No XORs' solutions discussed above on your breadboard to verify that they work as expected."

### BASICally speaking

I know we've been wandering around a bit recently, but now we are quivering in anticipation, poised to return to the next stage of implementing our retro games console, which is shown in Photo 1.

In our case, we are planning on controlling our console with an Arduino. However, as we've previously discussed, this processor will be presented in the form of a "game cartridge" that plugs into the console (the green board in the upper left of Photo 1).

This means that other users could build the same base console while implementing their games cartridges using different processors.

In fact, this is the case with the prototype shown in Photo 1, which was created by my friend Joe Farr. Joe is a big fan of PIC microcontrollers from Microchip, so that's the processor he used in Photo 1.

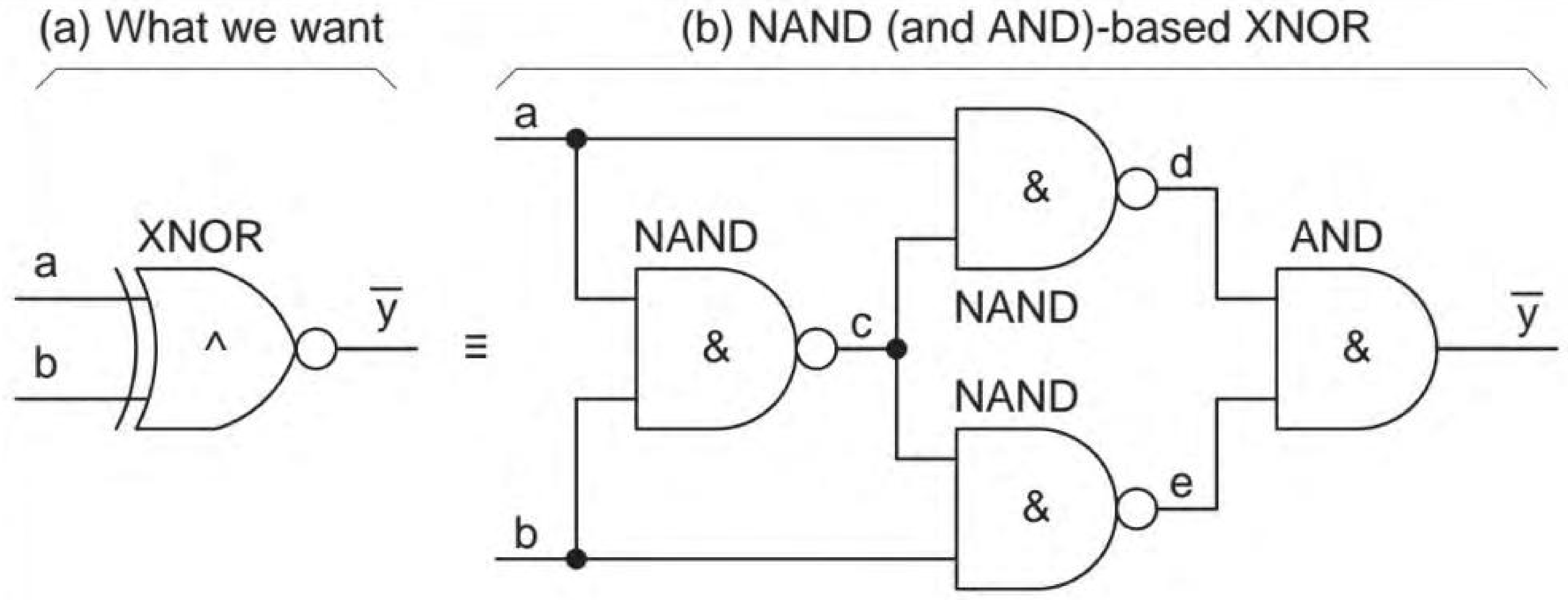
The reason I mention this here is on the off chance you use PICs yourself, or plan on doing so at some stage in the future. PICs offer a lot of advantages in terms of size and cost, but programming them can be a great big pain.

One solution I totally recommend involves the Positron compilers, which were developed by my friend Les Johnson. These are high-level compilers designed to make programming PIC microcontrollers easier. They use a language similar to BASIC and let developers write clear, structured code to control the microcontroller hardware.

The reason I wrote "compilers" (plural) is that there are two versions: the Positron8 for 8-bit PIC devices (like PIC16 and PIC18), and the Positron16 for 16-bit PIC devices (like PIC24 and dsPIC33).

Happily, this is all largely hidden from the user by the Positron IDE, which is called Positron Studio. You specify the PIC you are using in your code, and this causes the IDE to automatically employ the appropriate compiler. There's also a very active users forum where you can ask questions and share information.

All I can say is that Positron Compilers are fantastic value at £39.99 (https://pemag.au/link/ac70).



| a | b | $\overline{y}$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| a | b | c | d | e | $\overline{y}$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |

Fig.15: Implementing the XNOR function with three NAND gates and an AND gate.

I've no idea why they call me "Max Max"!

### Next time

In our next column, we are going to ramp up work on our retro games console, including wiring up the 7-segment displays and the eight pushbutton switches on the front panel. We will then consider what's necessary to control the whole thing with our Arduino board.

As always, if you have any thoughts you'd care to share on anything you've read here, please feel free to email me at max@clivemaxfield.com

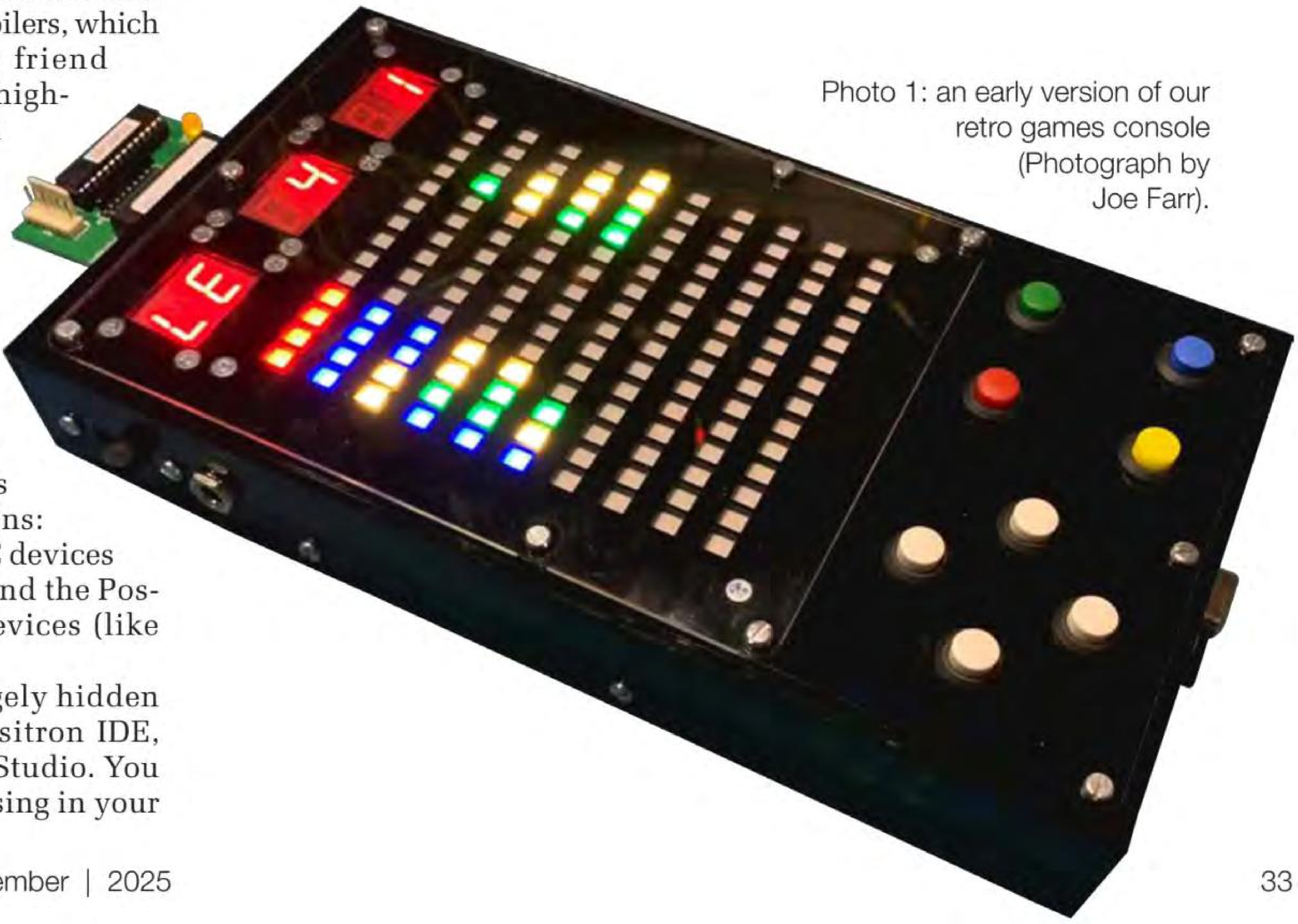


Photo 1: an early version of our retro games console (Photograph by Joe Farr).

Project By Charles Kosina

# AUTOMATIC LQ METER

INDUCTANCE / QUALITY

LQ METER
Q=236 @ 6.28MHz
L=5.9uH C=114pF
L
SET HIGH
START
ON
SET C

*A Q Meter is an indispensable tool for anyone contemplating RF design. My previous designs needed an external signal generator with a well-defined output level. This new design has a built-in signal generator and also features two instruments in one, measuring inductance from 0.1 to 999µH and Q from 10 to 300 with a test frequency from 100kHz to 90MHz!*

### Features & Specifications

- Measures inductance (L) and quality factor (Q) over five frequency ranges
- Inductance (L) range: 0.1-999µH with 100nH resolution
- Quality factor (Q) range: 10 to 300
- Test frequency range: 100kHz to 90MHz
- Resonant capacitance options: 18pF, 51pF, 118pF, 238pF or 488pF
- Power supply: battery (3 x AA) or 5V DC @ 200mA

**If you're working with inductors, it is helpful to understand** what we mean by the quality factor (Q) and how that affects the inductor's behaviour. In brief, an inductor with a low Q has more inherent damping, so it forms a filter with a broader response and a lower peak. In contrast, a high-Q inductor will make a filter with a narrow (more selective) response and a higher peak.

So you need to know the Q of the inductors in your filters, at the frequency they will operate, if you want to model their response accurately.

Besides adding the ability to measure inductance, so you don't need a separate LC meter, one of the big advantages of this design is that it has an onboard signal generator, so you don't need two instruments to make a Q measurement. Also, its operation is entirely automatic, whereas all my previous designs require fiddling with knobs and a specific procedure to make the measurement.

Much of the circuitry is similar to my older Q-meter designs, some of which were published in other magazines. Still, while I was adding the new features, I took the opportunity to optimise and simplify it without sacrificing any performance.

It appears that no manufacturers make Q meters any more, and the scarce second-hand ones from the likes of Hewlett-Packard fetch quite large sums.

I saw one recently selling on eBay for US$2400. This one costs a small fraction of that to build!

## Basic operation

Briefly, we can determine both the inductance and Q by exciting a resonant LC network containing the unknown inductor and a known capacitance at a controlled frequency. There will be a peak in the amplitude of the resonance at a particular frequency. The relevant formula is:

$$f = 1 \div (2\pi \times \sqrt{LC})$$

Since we know f and C, we can rearrange it to solve for L, giving us:

$$L = 1 \div C(2\pi f)^2$$

f is the resonant frequency, so we can sweep the oscillator and find the point at which the amplitude is at a maximum, then plug that into the formula. Changing C will shift the resonant frequency but should give us the same inductance result. That is necessary so that small and large inductance values can be measured at a reasonable frequency (within the device's operating range).

As for the Q factor, once we've found the peak, we can also measure the amplitude of resonance. The ratio between that and the excitation amplitude will give us our Q measurement, as we shall explain in a little more detail later.

## Design decisions

My first decision was how to generate the test signal over the required range. My first idea was to use a DDS chip such as the AD9851. However, with a clock frequency of 180MHz, the Nyquist limit is 90MHz, so 70MHz is about the highest frequency it can practically generate. Also, it's a relatively expensive chip or module.

Fellow project designer, Andrew Woodfield, suggested using the Silicon Labs Si5351 clock generator. I have used this chip in other applications, and it is extremely versatile, going up to 200MHz and beyond. These are available as ready-made modules with 25MHz crystals at a very low cost from AliExpress and other suppliers.

Its frequency is set by loading many registers over an I²C serial bus. That makes it easy for me to use a microcontroller to perform a continuous frequency scan.

The output of the Si5351 chip is buffered by a high-speed op amp, the OPA2890, configured with a unity gain. This has a gain bandwidth (GBW) of over 200MHz, so it will have a reasonably flat output to at least 90MHz. As with the previous design, the output of the OPA2890 feeds a toroidal transformer with a 10:1 turns ratio, the secondary being a threaded metal standoff passing through the middle.

This gives an extremely low source impedance to drive the series-tuned LC circuit, typically 0.02Ω. The voltage on the secondary is about 0.25V peak-to-peak. The catch is that the output is not a sinewave but more like a square wave. Instead of just one frequency, we have the Fourier expansion with an infinite number of odd harmonics:

$$\sin(\omega) + \sin(3\omega) \div 3 + \sin(5\omega) \div 5 + \sin(7\omega) \div 7 + \sin(9\omega) \div 9 + \sin(11\omega) \div 11 \ldots$$

Where ω is 2π times the frequency. It's an infinite series, but in practice, the higher harmonics are filtered out by the bandwidth-limited circuitry. Consider that the resonant frequency of inductor and capacitor (LC) circuit may be 15MHz. If we drive it with a 5MHz square wave, the third harmonic will resonate and give us a false reading.

Fortunately, this problem is easy to overcome. Instead of scanning upwards in frequency, we scan downwards from the highest frequency. As long as the highest frequency is above the resonant point of the tuned circuit, the scan will find the primary resonance frequency on the way down.

LQ METER
LQ Meter
Ver 3.0 Bat 4.7V
ON
SET HIGH
SET C
START

**When starting up to Automatic LQ Meter, the screen should display a message similar to the one shown. The lead image (opposite) shows the Meter measuring an air coil.**

For example, say we have an air-wound inductor of 6μH and a test capacitance of 118pF. The resonant frequency is 5.88MHz. If we set our starting frequency at 30MHz and scan down, no other resonances will be found until we reach 5.88MHz, as the first significant harmonic, the third, will only occur with a test signal of 1.96MHz (5.88MHz ÷ 3).

Given a close-to-zero source impedance, the Q value is obtained from the equation $Q = V_{OUT} \div V_{IN}$, where $V_{IN}$ is the voltage from the transformer, and $V_{OUT}$ is the voltage at the junction of the inductor and capacitor.

For a maximum Q reading of 300 and a test signal of 250mV peak-to-peak, $V_{OUT}$ would be 75V peak-to-peak. We need to measure the input and output voltages accurately, but it's impractical to measure $V_{IN}$ accurately on the transformer's secondary. However, we know the voltage on the primary is ten times that. My testing shows that the voltage ratio is close to 10:1 over the entire frequency range.

## Accuracy

Measuring Q accurately is not easy. The error budget includes several parameters, including the source impedance of the signal generator. While it is low, it is non-zero. RF voltage measurements are subject to errors and the peak frequency found may be slightly off. The stray capacitance on the circuit board may not exhibit a high enough Q, which will decrease the measured value slightly.

I compared the readings with that of my Meguro Q meter, and they were generally within 10%. Inductance measurements are likely to be within 5%. However, even the HP 4342A laboratory instrument can't guarantee a particularly high accuracy; it has a tolerance of ±7% on Q values up to 300.

## Circuit description

The resulting circuit is shown in Fig.1. MOD1 is the test signal generator and its output is buffered by IC1a and AC-coupled to transformer T1. The DUT (inductor) is connected across CON3 & CON4. It forms a resonant circuit in combination with one of the 33pF, 100pF, 220pF and 470pF capacitors switched in or out of the circuit by relays RLY1-RLY4 plus the stray PCB capacitance of around 18pF (or just the stray capacitance if RLY1-RLY4 are all off).

A half-wave precision rectifier built around the other half of the OPA2890 (IC1b) measures the amplitude of the $V_{TEST}$ signal (at pin 1 of IC1a). The output of this rectifier is the DC peak and IC5b buffers that voltage.

The gain of this buffer stage is set to 1.25, compensating for a slight amplitude reduction due to the rectifier. The DC voltage feeds the ADC7 input on the Arduino Nano module for measurement using its internal ADC (analog-to-digital converter).

At the same time, schottky diode D7 half-wave rectifies the voltage at the junction of the DUT and the

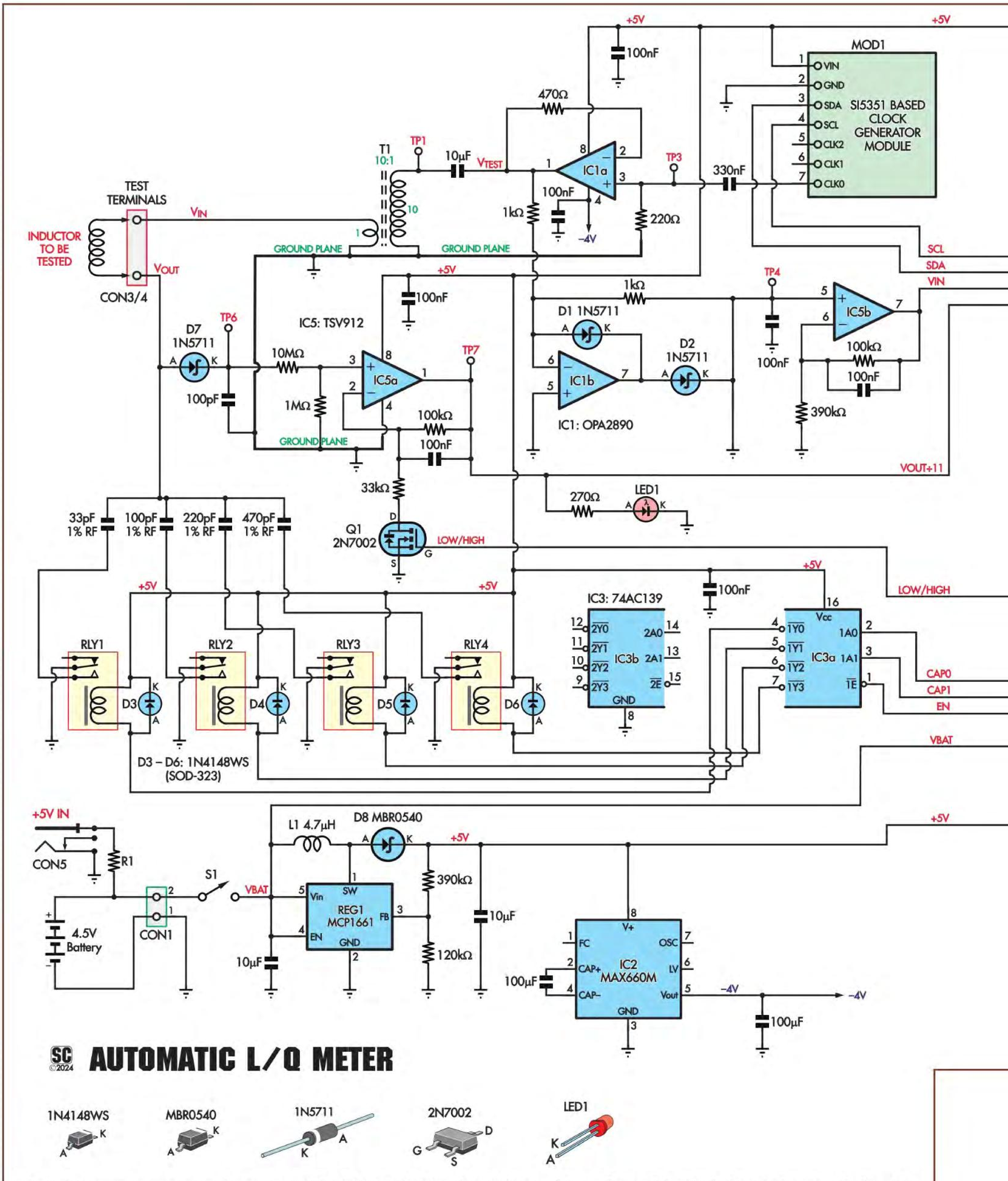
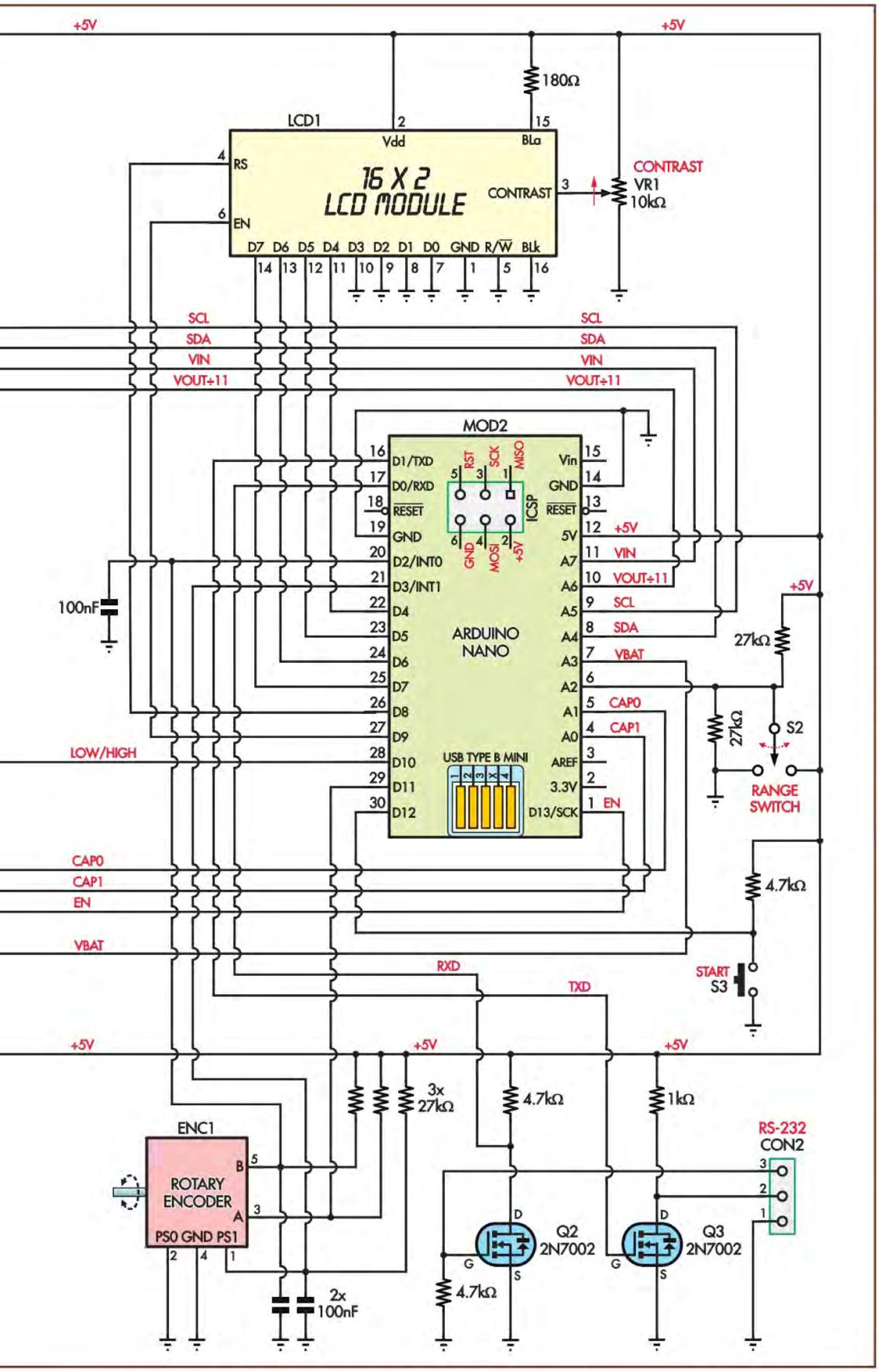

+5V
+5V
100nF
MOD1
VIN
GND
SDA
SCL
CLK2
CLK1
CLK0
SI5351 BASED CLOCK GENERATOR MODULE
470Ω
T1
10:1
TP1
10μF
VTEST
IC1a
TP3
330nF
100nF
1kΩ
220Ω
–4V
TEST TERMINALS
VIN
INDUCTOR TO BE TESTED
VOUT
CON3/4
GROUND PLANE
GROUND PLANE
SCL
SDA
VIN
+5V
100nF
1kΩ
TP4
IC5b
D1 1N5711
IC5: TSV912
D7 1N5711
TP6
10MΩ
IC5a
TP7
IC1b
D2 1N5711
100nF
100kΩ
100nF
390kΩ
100pF
1MΩ
100kΩ
100nF
IC1: OPA2890
GROUND PLANE
VOUT+11
33kΩ
270Ω
LED1
33pF 1% RF
100pF 1% RF
220pF 1% RF
470pF 1% RF
Q1 2N7002
LOW/HIGH
+5V
LOW/HIGH
+5V
+5V
100nF
IC3: 74AC139
RLY1
RLY2
RLY3
RLY4
D3
D4
D5
D6
2Y0
2Y1
2Y2
2Y3
2A0
2A1
2E
IC3b
GND
Vcc
1Y0
1Y1
1Y2
1Y3
1A0
1A1
1E
IC3a
CAP0
CAP1
EN
VBAT
D3 – D6: 1N4148WS (SOD-323)
+5V IN
CON5
R1
4.5V Battery
CON1
S1
VBAT
L1 4.7μH
D8 MBR0540
+5V
+5V
REG1 MCP1661
Vin
SW
EN
FB
GND
390kΩ
10μF
120kΩ
10μF
IC2 MAX660M
V+
FC
OSC
CAP+
LV
CAP–
Vout
GND
100μF
–4V
–4V
100μF
SC ©2024
AUTOMATIC L/Q METER
1N4148WS
MBR0540
1N5711
2N7002
LED1
A
K
G
D
S

**Fig.1: the test square wave is generated by MOD1, buffered by IC1a and transformed by T1 before being applied to the resonant circuit comprising the DUT and some combination of the 33pF, 100pF, 220pF & 470pF capacitors switched by RLY1-RLY4. The test and resonant voltages are rectified and measured by the Arduino Nano. By knowing the peak resonance frequency, capacitance and those voltages, both the inductance and Q factor can be calculated.**


+5V
LCD1
16 X 2 LCD MODULE
Vdd
BLa
RS
EN
CONTRAST
D7 D6 D5 D4 D3 D2 D1 D0 GND R/W BLk
180Ω
CONTRAST
VR1
10kΩ
SCL
SDA
VIN
VOUT÷11
MOD2
ARDUINO NANO
ICSP
RST SCK MISO
GND MOSI +5V
D1/TXD
D0/RXD
RESET
GND
D2/INT0
D3/INT1
D4
D5
D6
D7
D8
D9
D10
D11
D12
Vin
GND
RESET
5V
A7
A6
A5
A4
A3
A2
A1
A0
AREF
3.3V
D13/SCK
USB TYPE B MINI
VBAT
CAP0
CAP1
EN
100nF
27kΩ
27kΩ
S2
RANGE SWITCH
LOW/HIGH
4.7kΩ
START
S3
RXD
TXD
ENC1
ROTARY ENCODER
PS0 GND PS1
3x 27kΩ
2x 100nF
4.7kΩ
1kΩ
Q2 2N7002
Q3 2N7002
4.7kΩ
RS-232
CON2

capacitance ($V_{OUT}$), converting it to a DC voltage by charging a 100pF capacitor. A precision rectifier is unnecessary because the voltage here is much higher; a small voltage drop will not cause a significant error.

Applying a maximum of 37.5V DC to an op amp would destroy it, so we have an 11:1 voltage divider made from 10MΩ and 1MΩ resistors. This limits the output to 3.4V, which is a good safety margin. This divided voltage has a high source impedance, so IC5a buffers it before feeding it to the ADC6 (A6) analog input of the Nano.

The Nano's ADC has a resolution of ten bits, so the voltages at the ADC6 and ADC7 inputs are converted to an integral number from 0 to 1023 ($2^{10} - 1$).

The firmware calculation is simple: multiply the ADC6 value by 11 to recover $V_{OUT}$ and divide by the ADC7 value ($V_{IN}$). But what if we have a coil with a Q of only 10? $V_{OUT} \div 11$ would be only 0.225V, or an ADC count of 46, and the broad resonance peak may not be picked up accurately.

For low Q values, we increase the gain of IC5a from unity to four times by switching in a 33kΩ resistor from pin 2 to ground using N-channel Mosfet Q1. This will give an output voltage of 0.9V in this example, or 184 counts, which can be measured far more accurately.

## Resonant capacitance

In my original Q meter, I had eight capacitors switched by relays to select a value from 40pF to about 290pF with 1pF steps to move the frequency of the resonance peak. That was overkill, so I reduced it to a choice of only five values in this design. The stray capacitance of the circuit is around 18pF, setting the minimum value.

Why relays and not solid-state switching? To eliminate errors, the capacitance must have a very high Q, preferably ten times that of the highest Q coil. The relay contacts in series with the capacitors have very little effect on the overall Q.

The capacitors must be RF types with a 1% tolerance; the values are 33pF, 100pF, 220pF and 470pF, adding to the 18pF of stray capacitance.

## Power supply and control

Because op amp IC1a needs to drive the primary of T1 with a signal that swings above and below ground, its negative supply rail needs to be below 0V. We generate an approximately -4V supply rail from the +5V rail using IC7, a MAX660 switched capacitor voltage inverter in a fairly standard configuration.

The +5V rail is generated from a three-cell battery (at least 3V) by an MCP1661 switch-mode boost converter (REG1), again in a configuration pretty much straight out of the data sheet. This allows us to power the circuit with three AA or AAA cells (depending on how long we want them to last). The Nano can monitor the raw battery voltage via its ADC3 (A3) analog input.

Alternatively, 5V DC can be fed in from a USB supply, such as a phone charger, via CON5. In this case, REG1 will only operate to overcome the forward voltage of diode D8. If you use rechargeable cells (eg, NiMH), they will also be trickle-charged when external DC power is applied via R1.

The current drain in operation is about 200mA, so a decent set of AAs (alkaline or NiMH) should last for around ten hours of use. That might not seem very long, but this type of instrument is generally only used for a few minutes at a time, so the battery life should be OK unless you're using it constantly.

If battery operation is not needed, the MCP1661, the 4.7μH inductor and diode D8 may be omitted. Just put shorting links across the inductor and diode pads.

The rest of the circuit is pretty standard. The Arduino Nano has just enough I/O pins for the task. The LCD module is the standard 2x16 alphanumeric type available from multiple sources; the version with a blue backlight is recommended.

The four relays that switch the RF capacitors are selected by a 74AC139 multiplexer that will power the coil of just one relay at a time. The current sink capability of the 74AC139 is quite adequate for the relays used. Diodes across the relays absorb switching transients.

Rotary encoder ENC1 is a standard type with a 20mm-long shaft; 27kΩ pull-up resistors are used for the three switch contacts, with 100nF capacitors for debouncing on two of them.

Note that we have two capacitors on the INT0 line. One is located next to the encoder, but some noise spikes must have been getting into that line, making the frequency and capacitance settings erratic. A second capacitor right next to the Nano pin fixed the problem.

Two starting parameters can be set. The first is the top frequency, which can be set from 2MHz to 90MHz, while the other is the capacitance value to resonate with the inductor. Three-position switch S2 selects the setup mode. Up sets the top frequency, down sets the capacitor value and middle waits for the start switch (S3).

These additional switches also have pull-up resistors: 4.7kΩ for S3 and 27kΩ for S2. S2 feeds either 5V, 2.5V or 0V to the ADC2 (A2) pin of the Nano depending on its position, so an analog voltage measurement is used to determine its position.

## Finding the resonance peak

To find the peak voltage of the tuned circuit, we start at a high frequency and, at each step down, measure the voltage $V_{OUT}$. When far from resonance, this will be zero or close to zero. There may be a bit of noise, so the algorithm ignores anything less than an ADC count of 5.

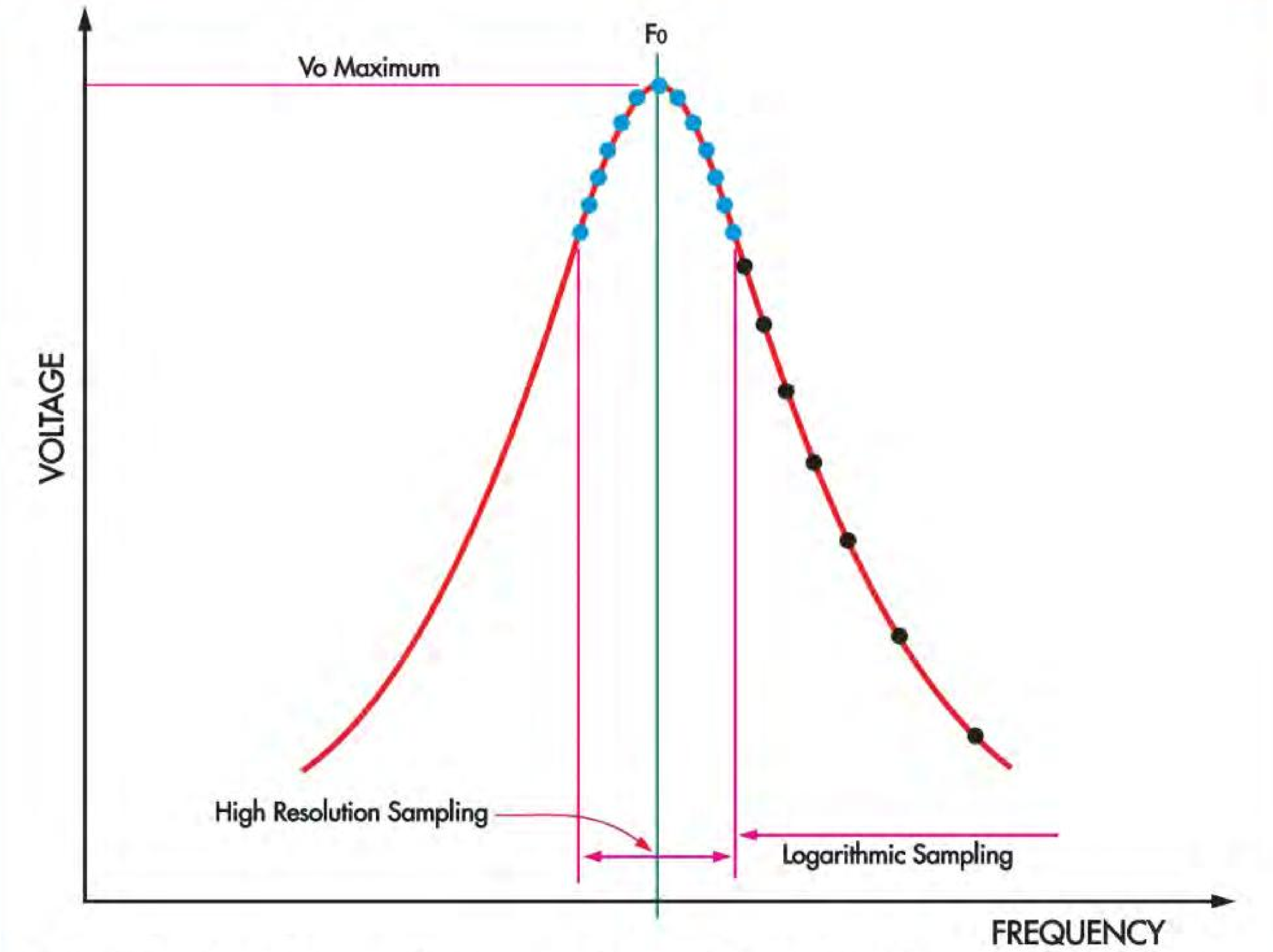


**Fig.2: this shows how voltage samples are taken at various widely-spaced frequencies until nearing the peak, at which point the unit switches to much smaller frequency steps. It's important to accurately find the peak frequency for precise measurements.**

The frequency steps far from resonance are at broad logarithmic intervals. That means that each step is the current frequency divided by a number. The logarithmic step size arrived at by experimentation is f ÷ 200. For example, at 10MHz, the next step size would be 50kHz (10MHz ÷ 200), making the next frequency 9.95MHz (10MHz – 50kHz).

The next step size would be 49.75kHz (9.95MHz ÷ 200) and so on.

When the measured voltage is 50 counts or greater on the ADC (about 250mV), we are on the rising side of the resonance curve, so we switch to a much smaller step size of f ÷ 4000. At each step, we measure the voltage and remember the highest voltage and the frequency at which it was found. If the voltage is lower than the highest seen so far, we increment a trailing-edge number instead.

When the trailing-edge number reaches five, we have passed the peak, so scanning stops. The highest stored voltage and frequency are then used to calculate the Q factor and the inductance. This is illustrated in Fig.2, where each point on the resonance curve is shown.

The peak will be sharp for high-Q circuits, so the sampling steps must be close together to avoid missing the peak.

During scanning, we switch to the low-Q setting by turning on Mosfet Q1 to increase the op amp's gain. This means that we will detect the rising slope sooner. If left on this setting, a high-Q coil could saturate the op amp output. To avoid that, we monitor the ADC count for $V_{OUT}$. If this exceeds 900, we switch Q1 off, reducing the measured $V_{OUT}$ by a factor of four.

As with the previous Q meter design, the brightness of LED1 is proportional to $V_{OUT}$. Because the algorithm takes the scan just past the peak, the LED will increase in brightness, dim slightly, then jump back to the highest brightness as we go back and re-measure the peak value.

Measuring RF voltages with great accuracy is not easy. Once the peak frequency is reached, both $V_{OUT}$ and $V_{IN}$ are sampled 16 times, and the readings are averaged. That helps to remove random noise.

I originally had some concerns about the accuracy of measurements due to the square wave shape. Is the rectified input voltage $V_{IN}$ different between a sinewave and a square wave? To test this, I used my previous Q meter and fed it with a sinewave and square wave generators. Over a frequency range of 1-10MHz, there was no significant difference in the measured Q.

## Construction

Most components mount on a double-sided circuit board coded CSE240203A that measures 138 × 75.5mm. The two modules, the Arduino Nano and the Si5351a clock generator board, are on the back of the PCB; almost all the remaining components are on the front.

Start by soldering in all the discrete resistors and capacitors in the locations shown in Fig.3, the PCB overlay diagram. As SMD capacitors do not have any markings, take care that the correct ones are soldered in. I use ceramic capacitors throughout, so like the resistors, their polarity does not matter.

Fit the SMD diodes next, all of which are polarised; their cathode stripes must be orientated as shown in Fig.3. The polarity of the surface mount diodes can be hard to see, so if you are unsure, test them with a multimeter.

Follow on by soldering the five integrated circuits, including REG4. None of them are particularly fine-pitch parts. Make sure that pin 1 is orientated correctly in each case, as fixing that after you've soldered all the pins is a chore!

The relays and 1N5711 axial diodes should be mounted next. Like the ICs, the relays must be orientated correctly. After that, solder the sole transistor (Q1) in place. Fit the 4.7µH inductor next; the SMD type is preferable for slightly higher efficiency.

It's a good idea to clean the PCB to remove flux residue before mounting the through-hole components, as it's easier at this stage. It's also a good idea to inspect all the SMD solder joints, especially for the ICs, before moving on, as it will be easier to fix any problems now.

## Winding the transformer

Wind ten turns of the specified enamelled copper wire onto the toroidal core (I used 0.4mm diameter wire but 0.25mm is OK), taking care that the turns are equally spaced around the circumference, to the extent possible,

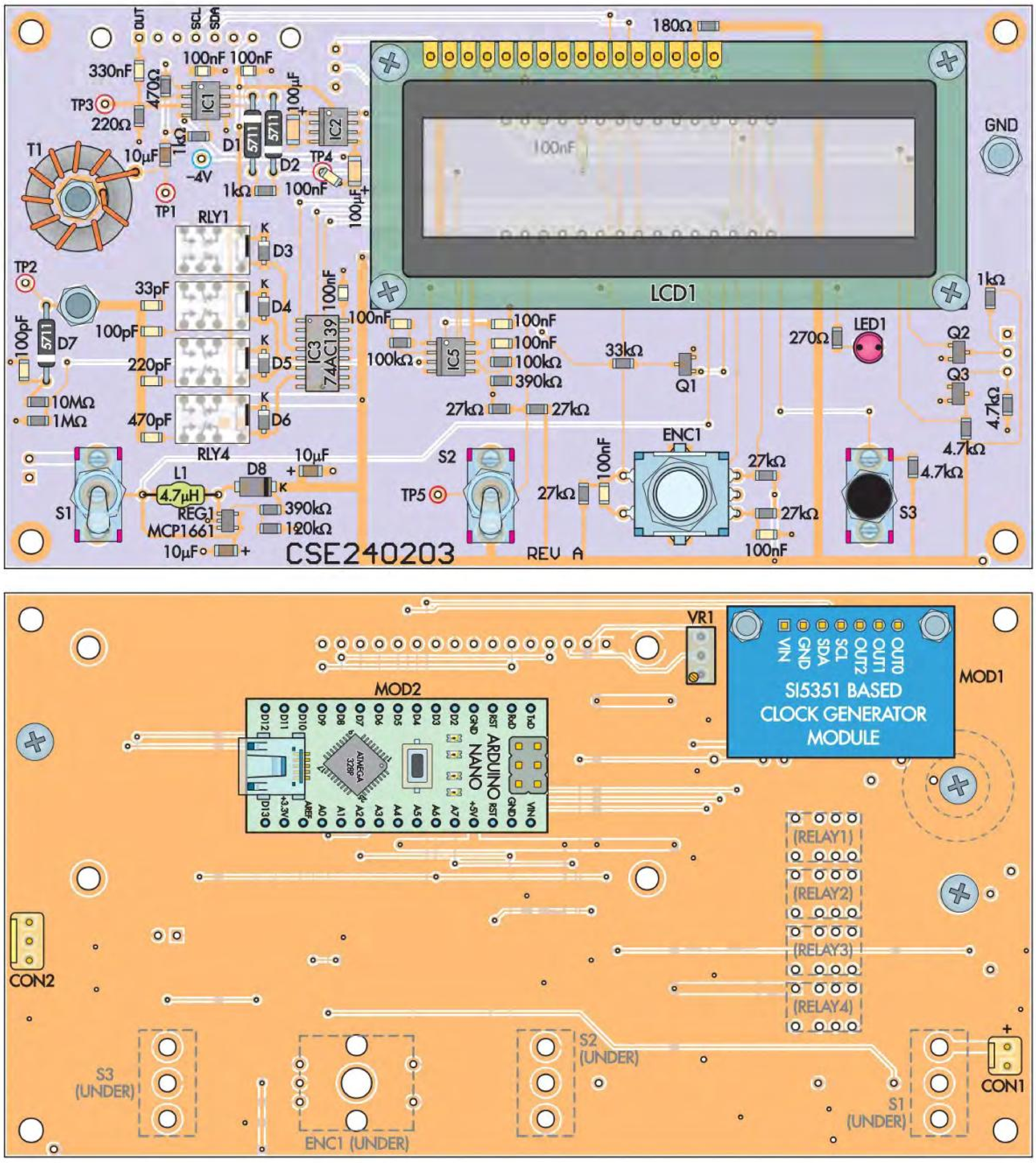
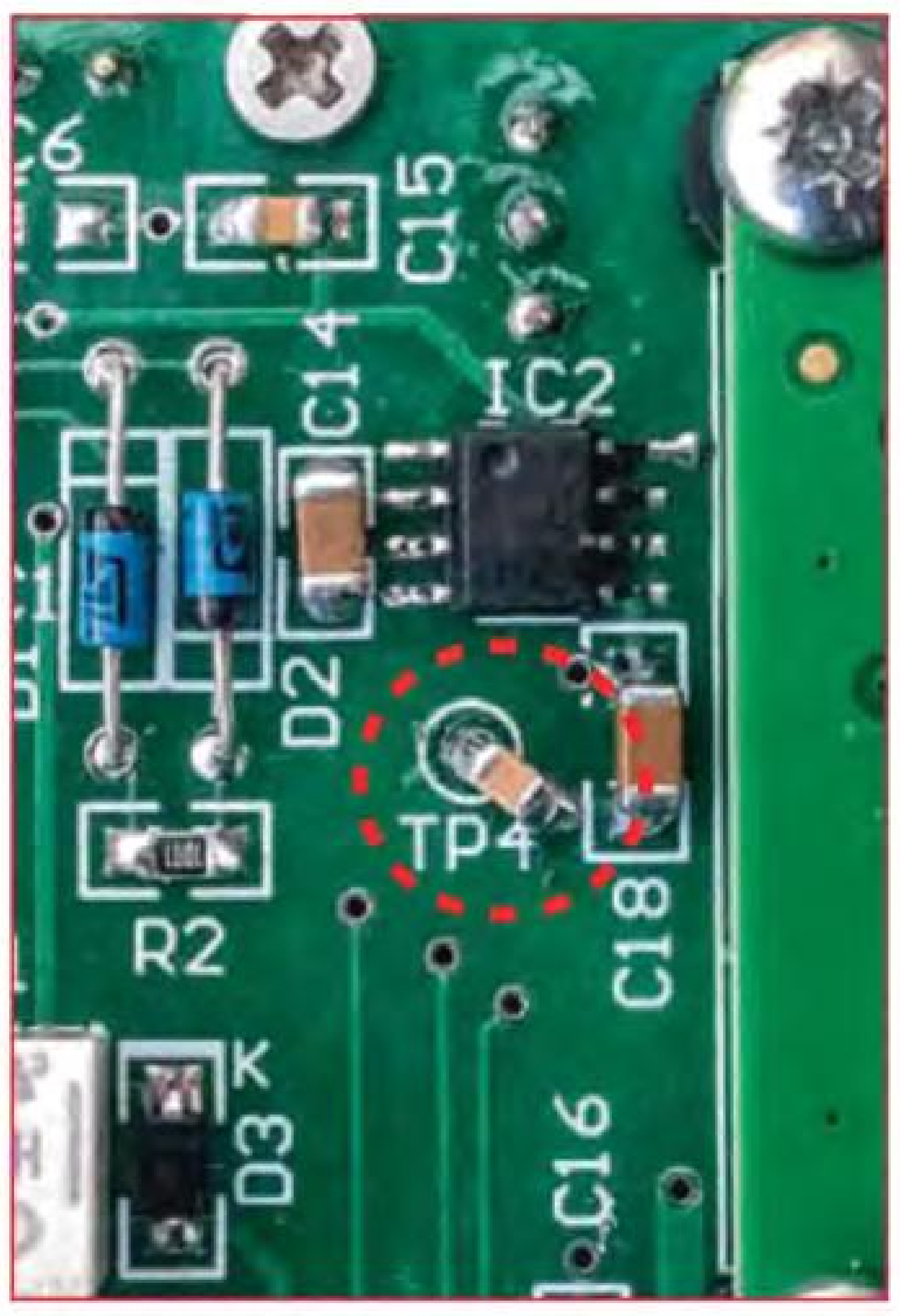

OUT
SCL
SDA
180Ω
330nF
100nF
100nF
470Ω
IC1
100µF
TP3
220Ω
IC2
GND
T1
10µF
1kΩ
D1
5711
5711
D2
TP4
100nF
-4V
1kΩ
100nF
TP1
100µF
RLY1
K
D3
TP2
33pF
100nF
K
D4
LCD1
1kΩ
100pF
5711
D7
100pF
IC3
74AC139
100nF
100nF
100nF
100kΩ
IC5
100kΩ
390kΩ
33kΩ
Q1
270Ω
LED1
Q2
Q3
220pF
K
D5
10MΩ
1MΩ
470pF
K
D6
27kΩ
27kΩ
4.7kΩ
4.7kΩ
4.7kΩ
RLY4
10µF
L1
4.7µH
D8
K
S2
TP5
ENC1
27kΩ
27kΩ
100nF
27kΩ
S1
REG1
MCP1661
390kΩ
120kΩ
S3
10µF
CSE240203
REV A
100nF
VR1
VIN
GND
SDA
SCL
OUT2
OUT1
OUT0
MOD1
SI5351 BASED
CLOCK GENERATOR
MODULE
MOD2
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
GND
RST
RX0
TX1
D13
+3.3V
AREF
A0
A1
A2
A3
A4
A5
A6
A7
+5V
RST
GND
VIN
ARDUINO
NANO
ATMEGA
328P
(RELAY1)
(RELAY2)
(RELAY3)
(RELAY4)
CON2
S3
(UNDER)
ENC1 (UNDER)
S2
(UNDER)
S1
(UNDER)
+
CON1

**Fig.3: most components mount on the top side of the PCB, with just the Arduino Nano, the Si5350a clock generator module and one or two headers on the underside. A large proportion of the parts are SMDs although they are almost all quite large and easy enough to work with. During assembly, take care with the orientations of the diodes, ICs and relays.**

***The top overlay diagram is the front of the PCB, while the bottom diagram is the back.***


C15
C14
IC2
D2
TP4
R2
C18
K
D3
C16

**The pads for one 100nF capacitor were accidentally left off the PCB, so it can be soldered like this (using a through-hole cap makes it easier).**

and that the ends line up with the two small pads on the PCB (one of which is attached to the large central hole).

Scrape the enamel off the ends of the wires, and tin them so they can be soldered to the PCB. Make sure it is centred correctly so that the spacer can pass through the middle.

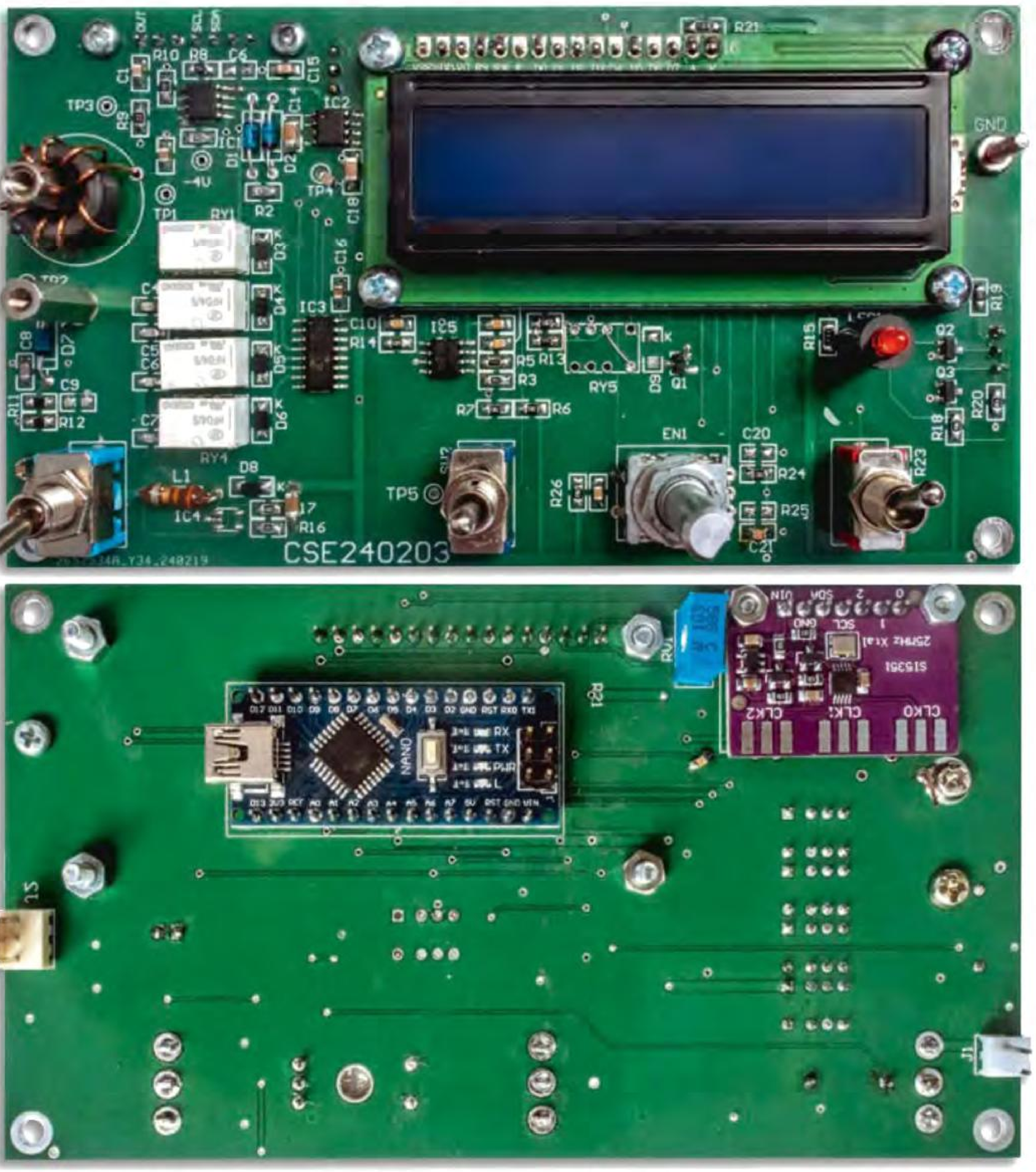
Once it is in place, gently feed one of the brass spacers through the hole in the middle of the toroidal core and feed in a bright metal M3 machine screw through the back of the PCB to attach it firmly (it needs to make good electrical contact). Attach the other brass spacer similarly to the hole just below the toroidal core and to the right of diode D7.

Now it's time to mount the various through-hole parts except the LCD, LED and modules. When fitting pushbutton switch S3, ensure that the NC contact goes towards the bottom of the board. Check which outer pin is connected to the middle pin with a continuity meter when the button is not being pushed; that is the NC contact.

Also take care that the switches and encoder are exactly at right angles to the board so that they fit through the front panel neatly. The best way to do this is to solder just one pin on each, then adjust their orientation so the front panel fits over them. Once you are happy with that, solder the remaining pins.

For the LED, insert its leads through the 8mm spacer before soldering it to the board. Its longer (anode) lead goes to the left, next to the adjacent resistor. The flat side of the lens should face to the right.

Before mounting the LCD screen, the Arduino Nano and Si5351 modules must be attached to the back. You could use socket strips to mount them, but it is not essential. In each case, if the module didn't come with a header soldered to it, fit one now.



**Note that the LCD screen is soldered to the PCB, as there isn't enough clearance to mount it on a socket.**

Finally, attach the LCD module on the front with 10mm M3 screws, hex nuts and 3mm spacers. The Si5351 module is also held in place with M2/M2.5 screws and 3mm spacers.

After cleaning the circuit board again, inspect all soldered joints and touch up any problems. The photographs show a prototype version of the board; the revised one has a few changes. Several components were not required and were removed from the artwork, while others were added.

## Programming the Nano

Before the LQ Meter can be tested, the ATmega238 microcontroller on the Arduino Nano module must be programmed. The modules generally come preprogrammed with a bootloader, with the correct fuse settings and a 16MHz onboard crystal, so you just need to load the LQ Meter specific firmware.

How you do that depends on what equipment you have. The simplest way is to plug the Nano into your computer using a suitable USB cable and upload the HEX file using free Windows software called AVRDUDESS (download it from **https://pemag.au/link/aaxh** or use the command-line version, avrdude, if you're running Linux or macOS).

Download the firmware from **https://siliconchip.au/Shop/6/416** then unzip it and extract the HEX file. Run AVRDUDESS and set the programmer to Arduino, select the Nano's USB serial port, a baud rate of 115,200 or 57,600 (depending on your Nano) and click "Detect". If it doesn't find the chip, adjust the settings and try again.

Once it does, go to the Flash window, open the HEX file for this project and click the program button. You should get a confirmation message, and that's it – the Nano is ready to go.

## Initial Testing

Don't install the board in the enclosure yet. With the Nano programmed, a battery or external power supply can be connected to the board. Leave the power switch off and briefly connect a multimeter on its high current range across the power switch. Around 200mA should flow. A much higher current than that could indicate a short on the board. If all is well, proceed to the next stage.

Switch it on and adjust potentiometer VR1 until the LCD screen image is legible. Switch it off and on again; the splash screen will show the version number and the battery voltage. After a couple of seconds, the following screen shows the capacitor value and top frequency. To adjust these, use the centre toggle switch and set the values with the encoder.

Once the values have been set, press the encoder switch to store the values in EEPROM, which are read on the next power-up.

It's possible that the encoder will work backwards. This depends on the specifics of your encoder and is quite unpredictable. If that happens, switch off the power, hold down the encoder switch and switch the power back on. The display will show "Toggling Direction". The direction bit is stored in EEPROM and will give correct operation from then on.

## Final assembly

Use the front panel PCB as a template for drilling holes in the front panel of the enclosure; Fig.4 shows the hole sizes. The panel is a snug fit in the detent, which makes for accurate drilling. Note that the spacers have clearance holes in the case so that they contact the pads on the back of the front panel.

With the front panel in the enclosure slot, attach the red and black terminal posts. Two nuts are used on the posts, one on the outside of the panel to make good contact with the pad, the other on the inside with the washer. Tighten them well to maintain a low resistance. The circuit board can then be slotted in and attached by two black 8mm-long M3 machine screws and the nuts on the switches.

Tighten the inside nuts on the switches right down for a correct fit. Push the knob onto the encoder shaft, and the unit is nearly complete. All that remains is to mount the battery holder and DC socket (for external power or battery charging) in the base of the case and wire them up.

Drill a hole in the side for the DC socket (if you're using it) and mount it. Make sure it won't foul the PCB or battery holder once it has been installed. Attach the battery holder to the base using double-sided tape, then solder the 47Ω axial resistor between the DC socket's positive terminal and the battery holder's positive wire. Solder the

## Parts List – Automatic LQ Meter

1 double-sided PCB coded CSE240203A, 138 × 75.5 × 1.6mm
1 double-sided front-panel PCB coded CSE240204A, black solder mask, 138.5 × 76 × 1mm
1 165 × 85 × 55mm IP65 sealed ABS enclosure with clear lid
**[Hammond RP1175C – RS 268-1717]**
1 Si5351A clock generator module (MOD1)
1 Arduino Nano (MOD2)
1 16×2 alphanumeric LCD with blue backlight (LCD1)
4 HFD4/5 subminiature DIP signal relays (RLY1-RLY4) **[AliExpress]**
1 Fair-rite 5943000301 ferrite toroid (T1) **[Farnell 2948713]**
1 30cm length of 0.25-0.4mm diameter enamelled copper wire (T1)
1 4.7μH M3216/1206 SMD inductor or axial RF inductor (L1) **[Murata LQM31PN4R7M00L]**
1 rotary encoder with integral switch and 20mm-long shaft (ENC1)
1 knob to suit ENC1
1 SPDT miniature two-position toggle switch with solder tags (S1)
1 SPDT miniature centre-off latching toggle switch with solder tags (S2)
1 SPDT miniature momentary pushbutton switch with solder tags (S3)
1 10kΩ top-adjust multi-turn trimpot (VR1)
1 3 × AA side-by-side battery holder with flying leads (BAT1)
1 2-pin vertical polarised header with matching plug and pins (CON1)
1 4mm red binding post (CON3)
1 4mm black binding post (CON4)
1 panel-mount DC barrel socket (CON5)

**Semiconductors**
1 OPA2890 dual 210MHz op amp, SOIC-8 (IC1)
1 MAX660M switched capacitor voltage inverter, SOIC-8 (IC2)
1 74AC139 dual two-to-four decoder/multiplexer, SOIC-16 (IC3)
1 MCP1661T-E/OT boost regulator, SOT-23-5 (REG1)
1 TSV912(A)ID dual rail-to-rail output op amp, SOIC-8 (IC5)
1 2N7002 N-channel Mosfet, SOT-23 (Q1)
1 3mm red LED (LED1)
3 1N5711 RF schottky diodes, DO-35 (D1, D2, D7)
4 1N4148WS SMD signal diodes, SOD-323 (D3-D6)
1 MBR0540 50V 0.5A SMD schottky diode, SOD-123 (D8)

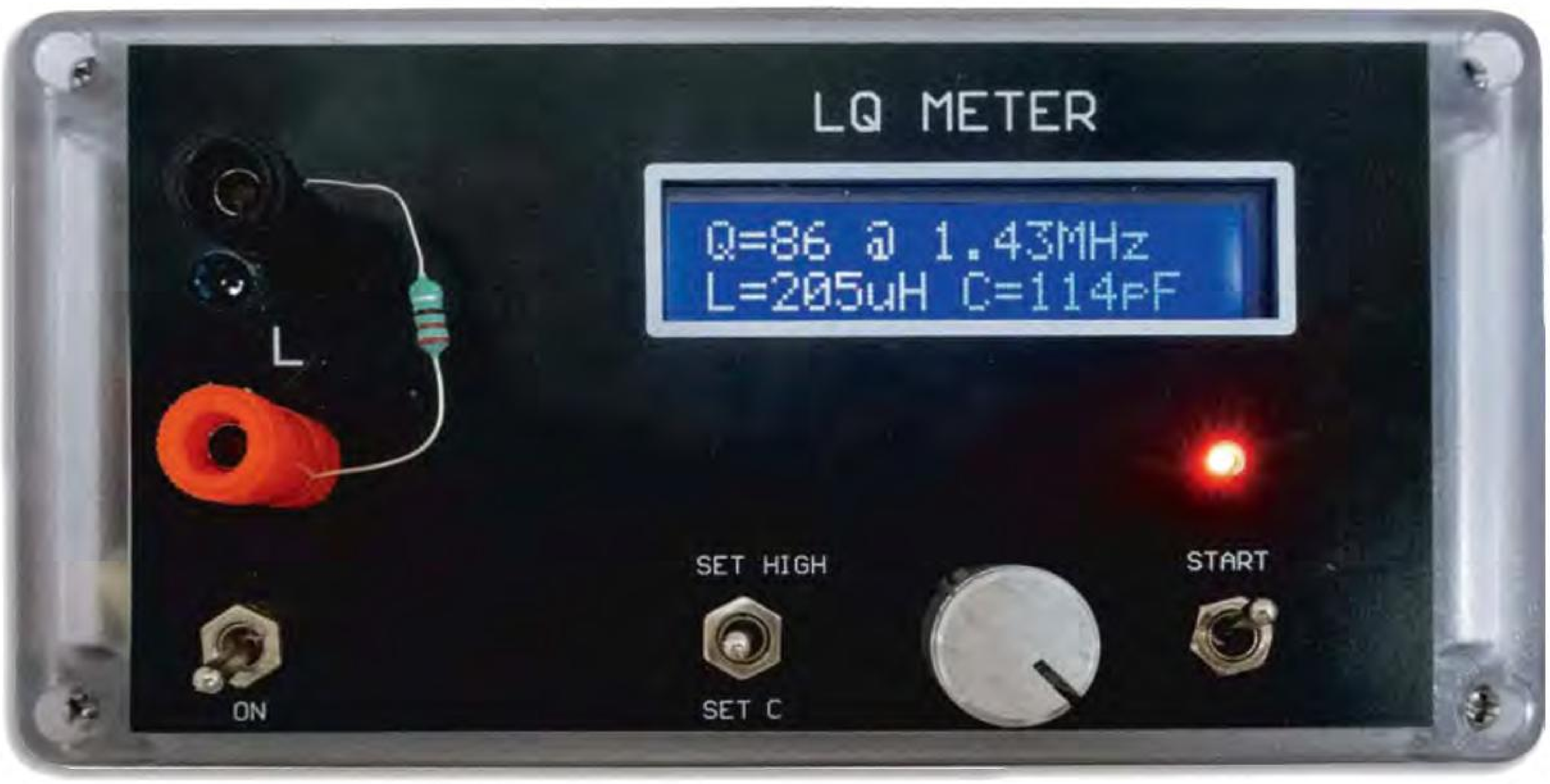
**Capacitors** *(all SMD M2012/0805 50V X7R 10% ceramic unless noted)*
2 100μF M3216/1206 6.3V X5R
3 10μF 6.3V X5R
1 330nF
10 100nF
1 470pF NP0/C0G RF (high-Q) 1%
1 220pF NP0/C0G RF (high-Q) 1%
2 100pF NP0/C0G 100V RF (high-Q) 1%
**[DigiKey KGQ21HCG2D101FT; Mouser 581-KGQ21HCG2A101FT; Farnell 1856269]**
1 33pF NP0/C0G 250V RF (high-Q) 1% **[Johanson 251R14S330JV4T]**

**Resistors** *(all SMD M2012/0805 1% unless noted)*
1 10MΩ
1 1MΩ
1 33kΩ
1 120kΩ
2 100kΩ
1 4.7kΩ
5 27kΩ
1 470Ω
1 270Ω
2 1kΩ
1 180Ω
1 47W ¼W axial (R1)
1 220Ω
2 390kΩ

**Hardware**
2 M3 × 16mm brass hex spacers
6 3mm ID 3mm-long untapped spacers
4 M3 × 10mm blackened panhead machine screws and hex nuts
2 M3 × 8mm blackened panhead machine screws
2 M3 × 8mm nickel-plated or stainless steel panhead machine screws
2 M2 × 10mm panhead machine screws and hex nuts
1 8mm-long LED spacer
1 double-sided foam-core tape pad approximately 40 × 60mm (for battery holder)
2 100mm lengths of light-duty or medium-duty hookup wire (red & black)

**Extra parts for optional debugging interface**
1 3-pin polarised header (CON2)
2 2N7002 N-channel Mosfets, SOT-23 (Q2 & Q3)
2 4.7kΩ SMD resistors, M2012/0805 1%
1 1kΩ SMD resistor, M2012/0805 1%

**SILICON CHIP Automatic LQ Meter Kits (SC6939, ~£50 + postage)**
*Includes everything in the parts list except the case & optional debugging parts.*



**The Automatic LQ Meter measuring a moulded inductor. You can rerun the test with different resonant capacitance values to get measurements at various frequencies.**

battery negative wire to the DC socket's ground tab.

You can find the positive tab on the DC socket using a continuity tester touching the central pin in the socket. It will make a sound when the other lead touches the correct tab. The ground tab is trickier since many sockets incorporate a ground switch; make sure a plug is inserted in the socket (but no power is applied) and check for continuity with the outer barrel of the plug and one of the tabs.

All that remains is to crimp (and possibly solder) two lengths of light-duty hookup wire into the polarised header plug and solder them in parallel with the battery leads.

Make sure that when it's plugged into the polarised header (CON1) on the PCB, ground goes to the bottom terminal and the positive supply to the upper terminal that connects to switch S1.

There is no reverse polarity protection on the PCB, so if you get this wrong, smoke will escape! Double-check that you got it right when the wires are connected by the PCB by verifying continuity from the battery's ground lead to one of the screw holes on the PCB and the outer barrel of the DC socket.

## Using it

Using the LQ meter is straightforward. Just connect the unknown inductor and press START. If you have no idea what the inductance is, set the frequency to the highest (90MHz) and the capacitor value to 51pF. It will take a few seconds to run its scan and display the Q and inductance values.

If you have a rough idea of the inductance, a lower top frequency will make the scanning faster. The calculation is according to the equation:

$$f = \sqrt{25330 \div LC}$$

... where f is the frequency in MHz, L is the inductance in µH and C is the capacitance in pF. The constant 25330 takes into account those units, plus the various gain or attenuation factors in the circuitry, as well as the ADC range.

The inductance of air-cored inductors will not vary much with frequency. However, the permeability of ferrite or iron cores varies with frequency, so you will get different values over the frequency range.

The five-capacitance range of this unit is comparable to the variable capacitor in Q meters of the past.

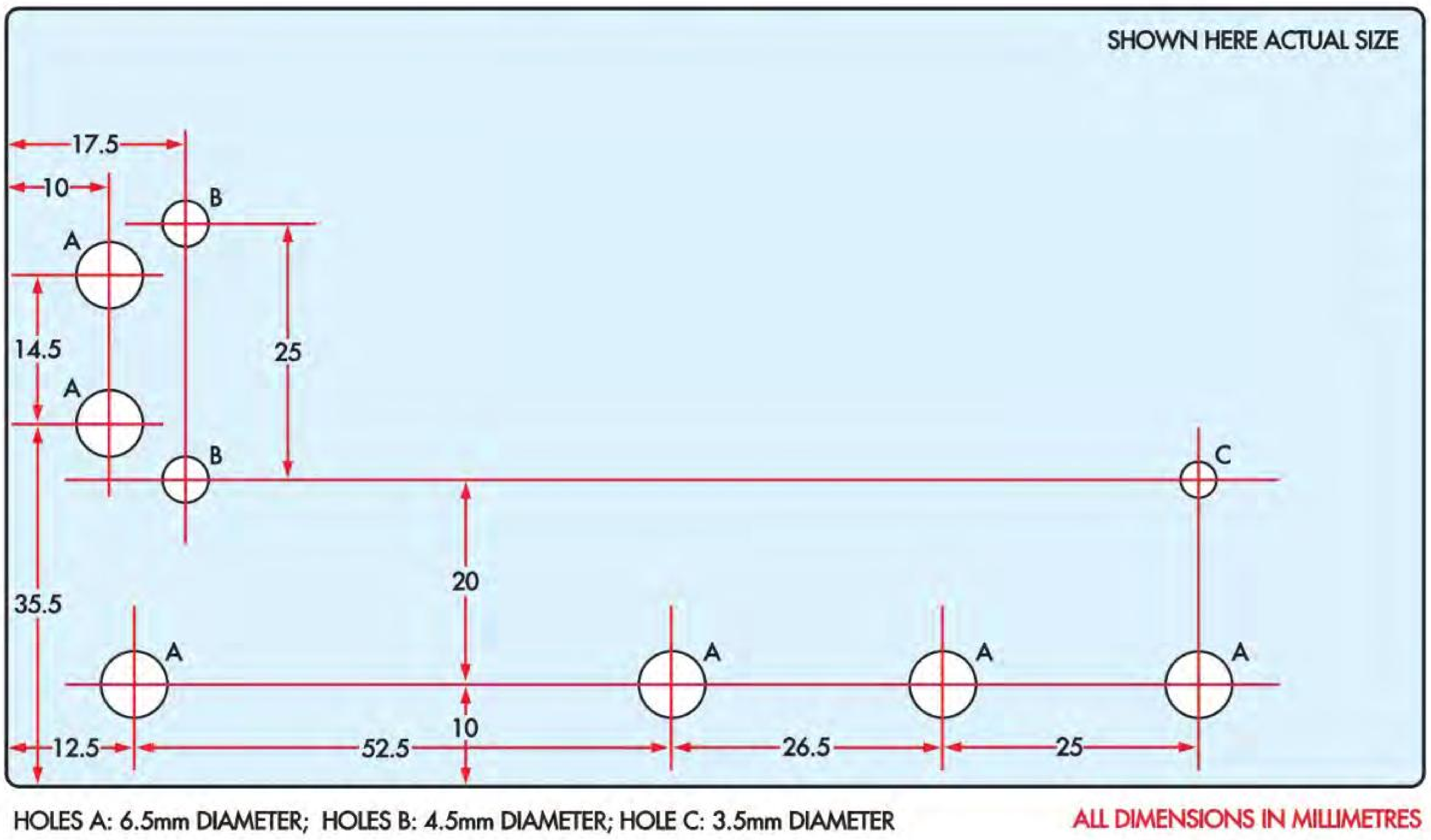

SHOWN HERE ACTUAL SIZE
17.5
10
B
A
14.5
A
25
B
C
20
35.5
A
A
A
A
10
12.5
52.5
26.5
25
HOLES A: 6.5mm DIAMETER; HOLES B: 4.5mm DIAMETER; HOLE C: 3.5mm DIAMETER
ALL DIMENSIONS IN MILLIMETRES

**Fig.4: use the front panel PCB as a template to drill holes in the front panel; they should be close to the positions shown here. Once they have been located with a pilot drill, enlarge them to the sizes shown here.**

# Practical Electronics

## UK readers SAVE £1 on every issue

# SUBSCRIBE NOW!

# 01202 087631

**Take out a one-year subscription and save more than £10 over the year.**
**Even better – save over £1 per issue if you subscribe for two years – a total saving of £26.**

## Overseas rates represent exceptional value

**You also:**
- **Avoid any cover price increase for the duration of your subscription**
- **Get your magazine delivered to your door each month**
- **Guarantee a copy, even if your local newsagent sells out**

**Order by:**
- **Phone or post with a cheque, postal order or credit card**
- **Or order online at: www.electronpublishing.com**

## Practical Electronics

## SUBSCRIPTION PRICES

Subscriptions for delivery direct to any address in:

**UK:** 6-months £33.99, 12-months £59.99, 24-months £114.99

**Europe Airmail:** 6-months £38.99, 12-months £69.99, 24-months £129.99

**Rest Of The World Airmail:** 6-months £46.99, 12-months £84.99, 24-months £164.99

Cheques (in **£ sterling only**) payable to ***Select Publisher Services Ltd*** and sent to: Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

**Tel:** 01202 087631
**Email:** pesubs@selectps.com

Also via our online shop at: **www.electronpublishing.com**

Subscriptions start with the next available issue. We accept MasterCard or Visa.
(For past issues see the Back Issues page.)

## DIGITAL SUBSCRIPTIONS

Electronic subscriptions are available from £29.99 for 12 months, more details at: **www.electronpublishing.com**

## SUBSCRIPTION ORDER FORM

☐ 6 Months: UK £33.99, Europe £38.99 (Airmail), Rest of the World £46.99 (Airmail)
☐ 1 Year: UK £59.99, Europe £69.99 (Airmail), Rest of the World £84.99 (Airmail)
☐ 2 Years: UK £114.99, Europe £129.99 (Airmail), Rest of the World £164.99 (Airmail)

**To:** **Practical Electronics Subscriptions**
**PO Box 6337**
**Bournemouth BH1 9EH**
**United Kingdom**

**Tel:** 01202 087631
**Email:** pesubs@selectps.com

**I enclose payment of £ .............**
**(cheque/PO in £ sterling only)**
payable to ***Select Publisher Services Ltd***

☐ Please charge my Visa/Mastercard

My card number is: ......................................................................
Please print clearly, and check that you have the number correct

Card Security Code ................. Valid From Date.......................
(The last 3 digits on or just under the signature strip)

Card Ex. Date ......................................

Name ..............................................................................

Address ...........................................................................

Post code ............................... Tel. ..........................................

Email ..............................................................................

**Subscriptions start with the next available issue.**

# Net Work

Alan Winstanley

The welcome return of my occasional column brings news of trends in smart meters, the onslaught of Chinese-made electric vehicles, and tempts readers into trying Temu.

**As promised in my April column,** I'm delighted to bring readers an occasional *Net Work* update, keeping them posted with topical trends and innovations in energy, electric vehicles, Internet usage, space programs and more besides.

Looking through my magazines archives, the February 1996 issue of *Everyday Practical Electronics* sparked some distant memories. This seminal issue introduced Derren Crome's Simple PIC Programmer, a small project built on Veroboard that empowered constructors to build feature-packed microcontroller-based projects like never before.

In the same issue was the late John Becker's PIC Electric Meter project, an advanced attempt at measuring the real-time electricity consumption of a single electrical appliance. Nowadays, cheap plug-in power meters do the same job. Still, with energy costs and financial pressures foremost in our minds, monitoring an entire household's energy consumption calls for a dedicated smart meter.

As *Net Work* readers know, smart meters have been one of the author's bugbears for a while now. Energy suppliers won't admit it, but smart meters are all about changing our behaviour to match an overstretched supply network with consumer demand. The faltering rollout of Britain's smart meters has seen some four million of them in circulation that do not work as advertised (if at all).

Problems with compatibility, network coverage and difficulties when changing suppliers have been widespread, but the country's £13.5 billion program continues unabated.

As I wrote in April, my supplier, EDF (Électricité de France), finally offered a £150 incentive to install a smart meter, but in the event I found that I'd have to upgrade my existing meter anyway. That's because my old meter had a certified lifespan of 20 years (2024) clearly stamped on it, and was therefore overdue for replacement.

In the same way, retail weigh scales, lorry weighbridges, beer and fuel pumps etc are legally 'stamped' and calibrated for trade use when selling goods by weight or volume. Otherwise, it's illegal to use them for trade.

My investigations into meter certifications turned up Schedule 7 of the 1989 Electricity Act, which gives suppliers the legal right to shut off the supply when non-approved meters are installed (more details are at https://pemag.au/link/ac7a).

Operating meters past their certified life makes them non-approved, so the supplier is legally obliged to swap them or risk prosecution. It's the one smart meter 'gotcha' that no-one can argue against.

## Not a bright spark

An electrician duly arrived and power was shut off while he rewired the electricity meter box. The matter of the In-Home Display (IHD) eventually cropped up. Strangely, my visiting 'sparky' tried to dissuade me from having one, saying, "You don't want one, do you?", adding that "most people don't like them because they don't like to be reminded of the cost".

More likely, I thought, it was the end of the day and he wanted to clock off and go home. So naturally, I insisted that an IHD was definitely required!

The LCD monitor took over 30 minutes to set up, with the disgruntled electrician muttering "this should be working by now" and "there's no signal" before it reluctantly sprang into life half an hour later.

Disappointingly, the IHD frequently loses contact with the smart meter and needs rebooting, sometimes several times a day, even when it's only 15 feet (about five metres) away from the meter itself. Ironically, the IHD is the most unreliable 'smart' network device that I've come across.

Even so, I admit that it's interesting to see how the household's power consumption racks up in real time, and how hidden 'phantom loads' silently clock up those kilowatt-hours, especially overnight.

I've chosen to transmit meter data once a day, although EDF pushes for 30-minute intervals.

The 'Loop' smart meter app for Android, iOS and Alexa is an alternative to using an In-Home Display. Loop fetches your smart meter data and displays it on a mobile device (also see *Net Work*, December 2022).

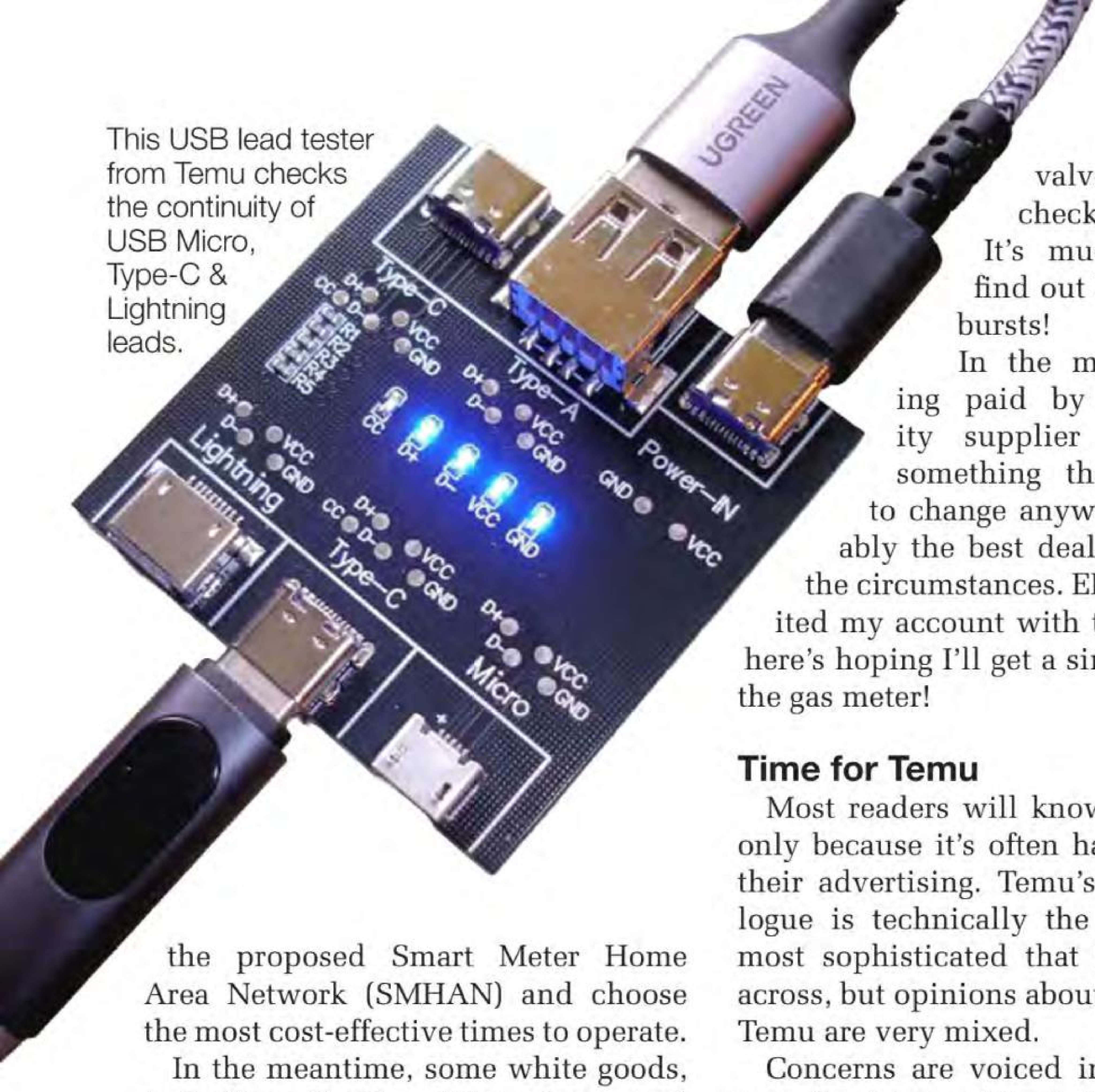
I decided to test it, so, after granting permission and verifying emails, Loop tried downloading my smart meter data, so far without success, perhaps due to network problems. I'll keep reporting back in future columns. Loop is free to use and has no sign-up charge. More details are at https://loop.homes/ and you can download the app from your usual source.

## Can't bash Bosch

As mentioned in April's column, Energy Smart Appliances (ESAs) promise in the future to communicate over

AUGUST 2004
Z20
M04
CERTIFIED
Single-Phase Watt Hour Meter
40316
1C
imp /
AMPY AUTOM
Property of: YFDL

An electricity meter stamped with a 20-year certification mark. The energy supplier must update it by law, inevitably to a smart meter.



This USB lead tester from Temu checks the continuity of USB Micro, Type-C & Lightning leads.

the proposed Smart Meter Home Area Network (SMHAN) and choose the most cost-effective times to operate.

In the meantime, some white goods, including Bosch dishwashers and washing machines, can also be managed through a useful app, with a 'Smart Start' function suggesting the best time to run, based on energy and time preferences.

I find the Bosch Home Connect app very useful for programming the dishwasher, checking water and energy consumption, and monitoring its progress. It even counts detergent tablets. I disabled the 'Allow remote start' network option though, as I like to physically switch the machine on in person.

It's clear how 'Demand Side Response' is gradually influencing us and controlling our energy consumption – it's supposedly to help us save money, but in reality it is to throttle demand for energy. I've written before that we can expect to see messages and promotions appearing on IHD screens as a matter of course, as the squeeze on our energy supplies continues.

The scourge of the smart meter has also spread to the author's water utilities. Unlike most other regions in Britain, my water authority fitted one at very short notice. Although there's no In-Home Display or app, a website login displays key data, showing a recent spike due to garden watering.

A plumbing mishap also revealed a potential problem: the water supply could not be isolated, as the meter had no key and only an enigmatic hex nut on top. It needed a 17mm AF socket – I sourced a spare on Temu (see later) for £2, just in case.

As an aside, I found the house's water stopcock had also seized solid, so I couldn't isolate the supply there either, until a plumber installed a new valve. It's worth checking yours. It's much better to find out before a pipe bursts!

In the meantime, being paid by an electricity supplier to upgrade something that they had to change anyway was probably the best deal going under the circumstances. EDF duly credited my account with the bonus, so here's hoping I'll get a similar deal for the gas meter!

## Time for Temu

Most readers will know of Temu, if only because it's often hard to escape their advertising. Temu's online catalogue is technically the slickest and most sophisticated that I have come across, but opinions about buying from Temu are very mixed.

Concerns are voiced in some quarters about toy safety or the supply of possibly dangerous, non-compliant or fake goods flooding into the country this way.

A USB rechargeable lamp I sourced from Temu was bundled with the obligatory cheap little USB-C lead for charging it up. I tried using a power bank without success, but when I unplugged the USB lead, I received a minor skin burn into the bargain! The lead was red hot and the insulation had melted.

I guess an internal short was shunting the power bank – another reason not to leave devices charging unattended. I now completely isolate cheap USB leads when not in use. A USB lead tester that I also sourced off Temu might detect such problems in the future. It needs a USB-C supply of its own, and will test Type-C, Micro and Lightning leads for continuity.

*[Editor's note – our USB Cable Tester project from the November & December 2022 issues runs from an internal battery. It checks cables and also reports on their DC resistance, to give you an idea of how much current they can handle.]*

Another lemon was a rechargeable colour-changing lamp, complete with remote control. After defying all attempts to charge it up, a twirl of a screwdriver revealed that the battery was missing altogether! Temu refunded the cost of this fake product immediately.

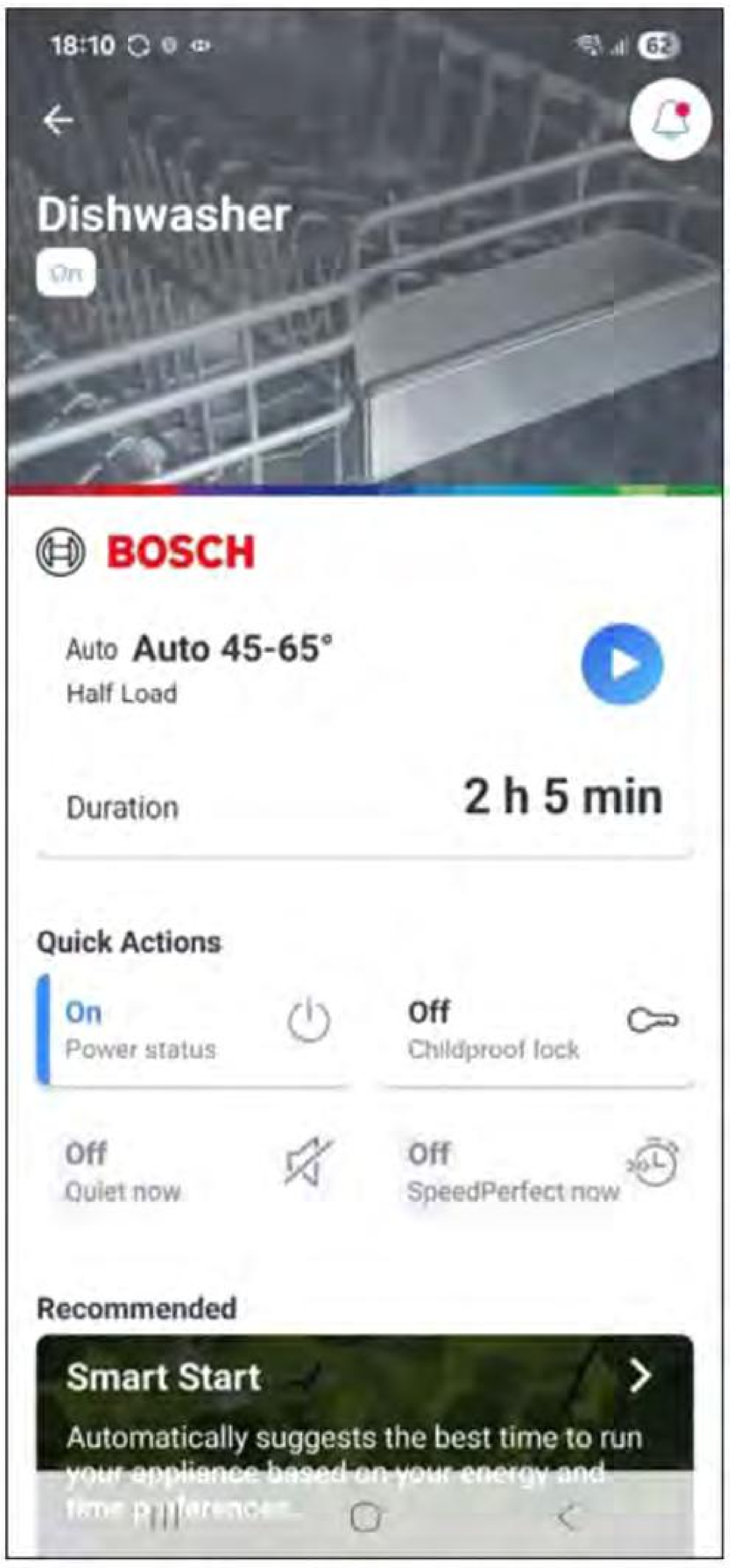
Such incidents offer a taste of what can go wrong in a minority of cases. Britain's consumer body, "Which?", has little positive to say about Temu, judging by a report last year (see https://www.which.co.uk/news/article/what-is-temu-and-is-it-safe-to-buy-from-a7BUb4k5fBFH).

Nevertheless, a huge variety of low-cost tools and garden accessories, handy do-dahs, accessories and merchandise for the crafter, home, garden, pets, wardrobe, kitchen and workshop are on show; tempting low-price offers may sway readers into buying.

For example, a useful air blower costing £10 has an 18650 battery and two nozzles that are clipped in place magnetically. It has four speeds (including 'violent wind gear', thankfully the maximum!) and is an ideal replacement for pricey (and chilly) flammable air duster aerosols.

For the engineer's workshop, there are selection boxes of fasteners, O-rings, R-clips, grommets, self-tappers, circlips, bolts, Nyloc nuts, hex screws of every size, and plenty more, ideal for keeping stocks topped up at very low cost. Reputable branded multimeters are also listed, including advanced large-screen LCD types, test prod sets and even 'talking' testers, all at affordable prices.

An expanding garden hose and trigger spray cost under £3 (special offer), a trivial consumable item but, more significantly, a trigger sprayer wall holder (£4) was 3D-printed and virtually indestructible.



Dishwasher

On

Auto Auto 45-65°
Half Load

Duration 2 h 5 min

Quick Actions

On Power status
Off Childproof lock
Off Quiet now
Off SpeedPerfect now

Recommended

Smart Start

Automatically suggests the best time to run your appliance based on your energy and time preferences.

The Bosch Home Connect app can operate compatible appliances such as dishwashers or washing machines, showing elapsed time, energy and water consumption.

This handy air blower from Temu starts at £10 and is safer than flammable aerosol dusters.

A tool that scores perforations in cardboard, to help when making up packaging materials, was also 3D-printed and cost just £3. A one-handed multimeter test probe clip caught my eye and cost just 61p. It's 3D-printed as well.

Elsewhere, a set of cleaning brushes, spotted on a TV shopping channel for £9.95 for two, cost 88 pence for five (true). A soft detailing/dusting brush, costing £7 elsewhere and ideal for cleaning electronics, ventilation slots, car dashboards or ornaments, costs just 39 pence each from Temu.

The list goes on; sachet wipes for removing sticky labels, and isopropanol wipes for cleaning spectacles or phone screens, cost about a penny each. I added a set of craft knives and separate packs of blades, plus a small receptacle with an internal magnet for snapping off and storing blunt blades safely.

I found an excellent large-display digital timer with a clock, 24-hour up/down timer and built-in alarm, for just £3.

This multimeter probe grip enables one-handed operation. It's 3D-printed and very cheap from Temu.

Temu's minimum order is £15, with free delivery. Merchandise arrives stuffed loosely into a polybag; larger, fragile or brittle items might not survive the journey. If so, Temu refunds immediately, and a £4 credit is sent if goods arrive late (which has happened to me twice).

I suspect that some Chinese vendors or drop-shippers who sell through Amazon Marketplace can't believe their luck when dealing with the British public. Identical cheap-and-cheerful goods that appear on Temu sometimes sell for ten times more on Amazon (eg, air blowers for £50-£80).

I often find it's worth comparing Amazon prices with Temu, while being wary of low-cost fakes and counterfeits that are usually easy to spot.

If you haven't tried Temu, then you might be pleasantly surprised by what you find – the more you look, the deeper it opens up. In my own experience, Temu's service has been acceptable enough, provided one applies some savvy and common sense. The vast range of affordable merchandise more than compensates for buying the odd dud or two.

My advice is to avoid signing up to notifications about deliveries and offers, though, or you will be pestered forever with offers for things you never knew you needed. Visit **www.temu.com** if you want to have a look, and the Temu app works impressively well, too.

## eBay Safety Gate

Fake or non-compliant products aren't unique to Temu or Amazon. In the July 2024 issue, I mentioned an adhesive called B-7000, a low-viscosity glue that is suitable for sealing plastic phone cases back together. I reckoned I'd keep a tube or two handy in the workshop.

Earlier this year, I received an alert from eBay, screaming that "There are product safety concerns with the item you purchased". It turned out that batches of Chinese-made B-7000, labelled with the CE mark, contained excessive levels (31%!) of toluene.

French analysts had reported the problem to the EU's 'Safety Gate' rapid alert system, so full marks must go to eBay for tracing and alerting buyers of the potentially dangerous batch of adhesive. The notice appeared online at **https://ec.europa.eu/safety-gate-alerts/screen/webReport/alertDetail/10008250**

eBay's new strategy of allowing private listings for free, while charging buyers a so-called "Buyer Protection Fee" instead, appears to be working smoothly. The options page for listing items is still onerous and time consuming to fill in and, annoyingly, eBay then tries to seize control and arrange shipping as well (I choose Royal Mail Tracked 48 instead).

Beware the hidden "Accept offers" option, enabled by default. Private sellers receive full payment a few days after a tracked delivery is confirmed; otherwise, payments are made by eBay a fortnight later. Sellers also now receive positive feedback from eBay automatically if buyers don't bother doing so.

## Building Your Dreams

*Net Work* has covered the steady march of electric vehicles ever since Chinese-owned MG cars first appeared on our roads. Many early adopters of expensive EVs love their cars, but some have had their fingers burnt when it comes to insuring, repairing or trading them in. Never mind facing the practicalities of living with them and charging them up.

Battery technology itself continues to evolve. BYD launched its "Super e-Platform" in China in March, promising

The BYD Shenzhen is the largest car transporter to date, carrying 9000 vehicles.

LNG POWERED
BYD
BYD SHENZHEN

MORE MAGAZINES:
XSAVA.XYZ

to deliver a 400km range with just five minutes of charging at special 'flash' charging stations, a similar refuelling turnaround of a petrol car. Whether Britain's faltering EV charging infrastructure will ever be up to the task of delivering energy that fast (around 1MW per vehicle!) is another story.

I've previously showcased what Chinese EV makers are now producing to highlight what's heading our way, though some early efforts like the Ora Funky Cat (*Net Work*, January 2022) and Fisker Ocean (*Net Work*, May 2022) failed to catch on. China dubs such cars 'New Energy Vehicles' (NEVs) and some exceptionally advanced design concepts are now on offer.

Chinese NEV manufacturer Chery's website is typical of how dazzling electronics, touchscreens and LEDs dominate a sales pitch, as shown at **https://www.cheryinternational.com/pc/models/tiggo/tiggo8promaxt18fl3/index.shtml**

Data from the SMMT, the UK motor trade body, shows how China's BYD is chipping away steadily at the British market (even my local Honda dealer has signed up to sell BYD), taking 1.74% of the UK market year-to-date, about the same as Tesla (1.76%).

More Chinese brands have recently arrived on our shores out of nowhere, as I forecast in previous *Net Work* columns. Newcomers include Chery's blunt-looking Jaecoo, which has already grabbed 0.7% of the market, while Chery's sub-brand Omoda also sold 0.6% according to the SMMT. Between them, these sales figures match the total sales of Honda cars this year.

In a tough market, prices are sharpening all the time, and the £26,000 petrol Omoda 5 was recently named Best Value New Car 2025 at the Autotrader Drivers' Choice Awards. Their Car of the Year is the electric Kia EV3 (see **https://www.autotrader.co.uk/cars/new/awards**).

More new arrivals include XPeng (*Net Work*, May 2023), selling just 51 cars, and Skywell (*Net Work*, March 2024), which registered exactly nine vehicles. Well, you have to start somewhere. It's a sign of how the Chinese automotive industry is hungry to gain a foothold abroad. In case you wondered, Volkswagen leads the way, with 9% of UK sales this year.

Following the political upheaval earlier this year, Tesla has fallen on slightly harder times. In May, it's reported that BYD outsold Tesla in Europe for the first time. BYD also easily outsold Tesla in China by 4.5 times in one week alone.

The new Xiaomi YU7 premium EV attracted 200,000 orders in just three minutes in China.

This year, BYD is building the world's largest factory complex in Zhengzhou, covering 130km$^2$ (32,000 acres or 50 square miles), and is limbering up to deliver its cars worldwide.

## Any port in a storm

Adding to its fleet of three ships, the world's largest car transporter vessel, the BYD Shenzhen, recently embarked on its maiden voyage and delivered 7000 EVs (capacity 9200) to Brazil, "showcasing [BYD's] dominance in the EV market and its commitment to sustainable transport solutions", said China Global TV.

Both BYD and Tesla face plenty of competition, though. At the time of writing, Xiaomi, better known in Britain for mobile phones and smart products, has just launched the Xiaomi YU7, a premium EV that 'broke the Internet' when it was released, says one Chinese news source.

The model is gunning for the Tesla Model Y. Reportedly, some 200,000 orders for the new SUV were placed in **three minutes**. In comparison, UK car sales total about 170,000 vehicles per month. The YU7 will cost $35,000-45,000 (£25,500-33,000), and Xiaomi has more at **https://www.mi.com/global/discover/article?id=4925**

The domestic Chinese car market has, however, been alarmed to discover a new ploy that distorts sales figures in a way that's been common in Britain for 30 years. Car dealers may 'buy' or register 'zero kilometre cars' themselves, just to inflate their sales figures, before reselling them as 'pre-registered' but otherwise brand new cars, at significantly lower prices.

The SMMT never replied when I asked whether their vehicle registration figures included pre-registered dealer figures.

There is much over-capacity in China's car industry, and the highly volatile EV market will only see more electric vehicles appearing on our roads as brands jostle to grab their share of the overseas market. The thorny issues of tariffs, trade deals and anti-dumping will also doubtless come to a head in the coming months.

## Frank W. Hyde on film

Following on the heels of SpaceX, Japan's Honda recently launched an experimental research rocket as it seeks to enter the reusable rocket market. Being a Honda, the rocket started first time, set off successfully and returned reliably to Earth without incident. The test flight can be seen on the NHK Japan website at **https://www3.nhk.or.jp/nhkworld/en/news/special/04/4086/**

Still on the topic of reaching for the stars, long-time readers of *Practical Electronics* will remember the popular Spacewatch column written by the late Frank W. Hyde. In *Net Work*, April 2023, I wrote how Frank Hyde built the world's largest amateur radio observatory in Britain in the 1960s.

He collaborated with the Jodrell Bank observatory, and it's said his work influenced some experiments onboard NASA's Pioneer missions. He also appeared on BBC TV's "Sky At Night" with British astronomer Patrick Moore.

I've managed to capture contemporary footage of Frank Hyde at work in his observatory, taken from a 1960s 'short' that was produced by the Rank Organisation for its very popular "Look at Life" series. I've uploaded the two-minute clip onto YouTube, at **https://youtu.be/nS8c1Is21vQ** and I'm sure regular *PE* readers will enjoy watching it.

That's all for this month. Readers will find all the above links ready-made on the *Net Work* blog at **electronpublishing.com** – join me next time for more topical news of developments in energy, space programs and getting the best out of surfing online. As always, you can email me at **alan@epemag.net** PE

The Frank W. Hyde clip now on YouTube.

# Circuit Surgery

Regular clinic by Ian Bell

## Topics in digital signal processing – Implementing DSP on a microcontroller, part four

**We are looking at various** topics related to digital signal processing (DSP).

DSP covers a wide range of electronics applications where signals are manipulated, analysed, generated, stored or displayed as digital data but originate from and/or are converted to real-world signals for interaction with humans or other parts of the physical world.

Fig.1 shows the key elements of a generic DSP system with a signal path from an analog input via digital processing to an analog output.

This does not necessarily represent every DSP system (not all have all the parts shown), but it serves as a reference for the various subsystems we will look at.

In the previous few articles, we have been working through an example microcontroller-based DSP implementation. Before looking at the implementation of a sinc filter, we discussed individual aspects of the system: the DAC (digital-to-analog converter) and ADC (analog-to-digital converter), sample timing and data transfer.

Last month, we discussed the ADC on the microcontroller we are using, and developed code to transfer the ADC readings to the DAC. This code creates a replica of the ADC input waveform on the DAC output. Initially, we used a timer-driven interrupt routine to transfer data, but this proved to be slow, running at a maximum sampling rate of around 60kHz.

Far more efficient data transfer is achieved using direct memory access (DMA), in which independent hardware (the DMA controller) moves data between the microcontroller's memory and peripherals such as the ADC and DAC. We introduced the basics of DMA and used it to implement ADC to DAC transfer.

This requires two buffers (contiguous areas of memory) to hold data read from the ADC and to be written to the DAC. The buffers are 'circular'; when the read operation reaches the upper address limit, it continues from the lower end of the buffer's address range.

The ADC-to-DAC transfer code simply has to copy data from the ADC buffer to the DAC buffer, coordinated with the DMA transfer timing. The microcontroller's software library provides two callback functions that run when the buffer is half-full and full.

Copying the data in these two functions means that code operations take place on stable data in one half of the buffer while the other half is being accessed by the DMA controller.

We concluded last month with the code shown in Listing 1. This approach provided a significant improvement in speed – it could run far faster than the DAC's maximum conversion rate of around 330kHz.

### FIR filter recap

This month we will look at the implementation of a finite impulse response (FIR) filter, specifically a windowed sinc digital filter.

We discussed digital filters in detail from December 2024 to May 2025, focusing on the theory, finding coefficient values and simulating the filters in LTspice.

As previously mentioned, our aim here is to develop a basic implementation of the filter, without attempting to use any advanced algorithmic or coding techniques. We will not use library code such as Arm's Common Microcontroller Software Interface Standard (CMSIS) DSP library so that the processing operations are not hidden in calls to library functions.

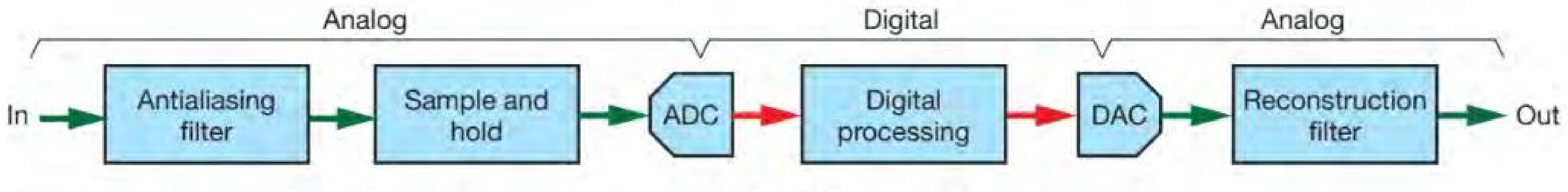


Fig.1: a generic digital signal processing (DSP) system structure.

The structure of an FIR filter is shown in Fig.2; its output is obtained from a weighted sum of the present and past samples; the weights are the coefficients (*a*) of the filter (collectively called the kernel). The samples are written as a set of indexed values with index *n* (input $x[n]$, output $y[n]$) referring to the most recent sample.

The previous input is $x[n-1]$, the one before that $x[n-2]$ and so on. In a software implementation of a filter, the samples are stored in system memory. Storing the previous sample and using it at the current time is equivalent to passing if through a delay (Δt) of one sample period (T).

For a filter with *N* coefficients, the output value is calculated from the present input sample and past (delayed) $N-1$ inputs. The input delayed by *i* samples ($x[n-i]$) is scaled by the coefficient $a[i]$. Mathematically, we are performing a convolution of the kernel and input signal (see *Circuit Surgery*, August 2024), which can be written as the summation (Σ) of $a_i x[n-i]$ for *i* from 0 to $N-1$, ie:

$$y[n]=\sum_{i=0}^{N-1} a[i]\,x[n-i]$$

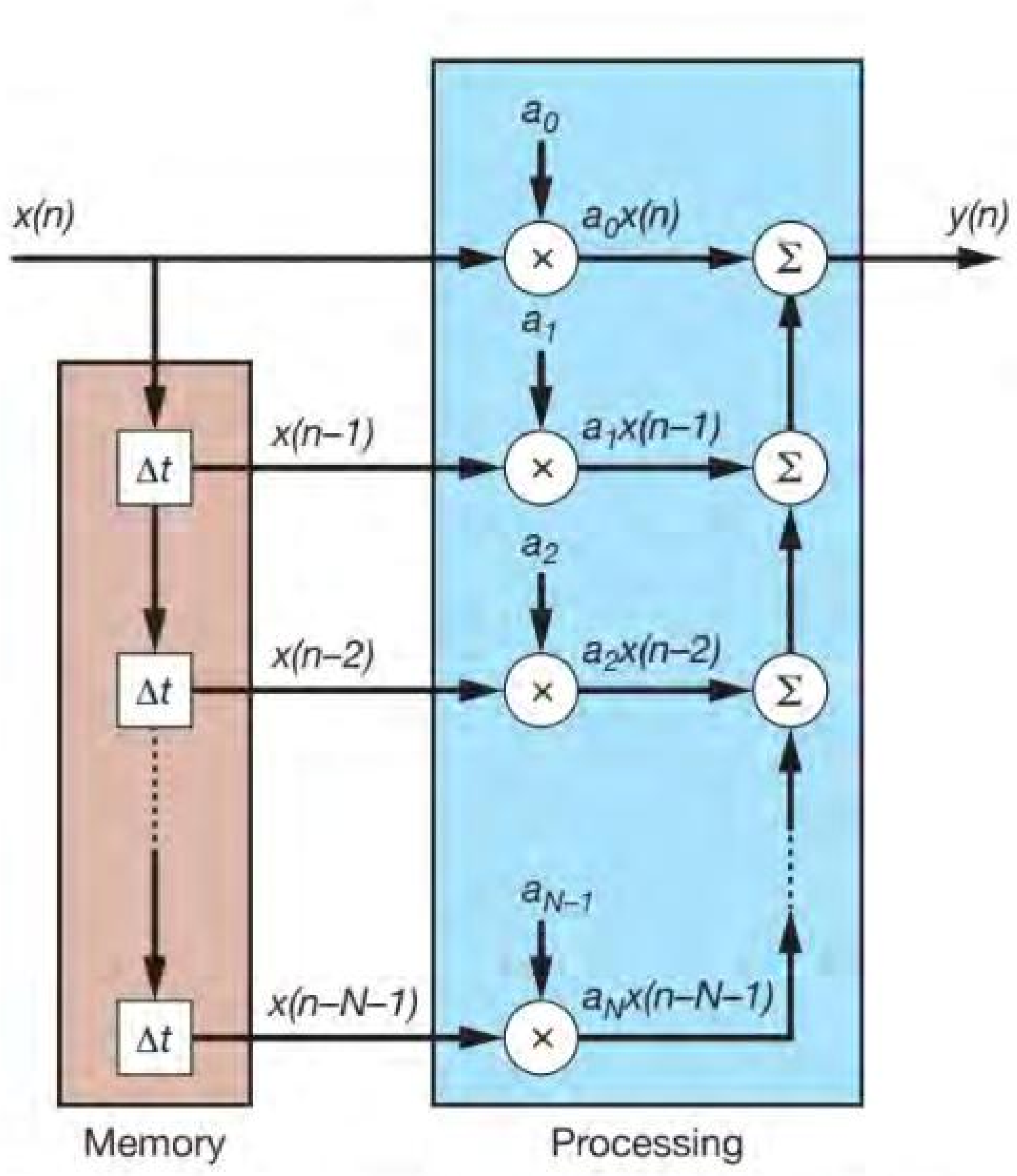


Fig.2: the structure of a finite impulse response (FIR) filter.

```
/* USER CODE BEGIN 4 */
void HAL_ADC_ConvHalfCpltCallback(ADC_HandleTypeDef* hadc)
{
  HAL_GPIO_WritePin(GPIOB, ARD_D8_Pin, GPIO_PIN_SET);
  for (int i= 0; i < HALF_BUF_SIZE; i++)
  {
    DAC_buffer[i] = ADC_buffer[i];
  }
  HAL_GPIO_WritePin(GPIOB, ARD_D8_Pin, GPIO_PIN_RESET);
}

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
  HAL_GPIO_WritePin(GPIOB, ARD_D8_Pin, GPIO_PIN_SET);
  for (int i= HALF_BUF_SIZE; i < BUF_SIZE; i++)
  {
    DAC_buffer[i] = ADC_buffer[i];
  }
  HAL_GPIO_WritePin(GPIOB, ARD_D8_Pin, GPIO_PIN_RESET);
}
/* USER CODE END 4 */
```

Listing 1: the DMA callback functions we concluded with last month.

```
Result = 0;
for (int i = 0; i < COEFF_SIZE; i++)
{
  Result = Result + Coefficient[i] * filter_input[n-i];
}
```

Listing 2: the FIR filter calculation for one output sample.

```
1  for (int i = 0; i < HALF_BUF_SIZE; i++)
2  {
3    filter_input[i] = (float) ADC_buffer[i];
4  }

5  for (int i = HALF_BUF_SIZE; i < BUF_SIZE; i++)
6  {
7    filter_input[i] = (float) ADC_buffer[i];
8  }
```

Listing 3: code snippets for converting and copying DAC data to the filter input buffer. Lines 1-4 are for the lower half of the buffer, with lines 5-8 for the upper half.

For example, for four coefficients, the FIR filter output for $y[n]$ is given by:

```
y[n] = a[0] × x[n]   +
       a[1] × x[n-1] +
       a[2] × x[n-2] +
       a[3] × x[n-3]
```

There are normally more coefficients, but I wanted to keep this example simple.

## Basic filter code

Coding the filter calculation in basic form is straightforward. If we assume that we have the coefficients in a floating-point array called **`Coefficient`** and the input samples in a floating-point array called **`filter_input`**, the code snippet in Listing 2 performs the calculation for $y[n]$, where $n$ is the index of the **`filter_input`** array corresponding with the set of input samples ($x[\,]$) in the equation above.

The code is generalised for any number of coefficients using the #defined value **`COEFF_SIZE`** value. The result is a floating-point number holding the accumulated summation. After the loop, the value of **`Result`** is the output sample value for sample $n$ ($y[n]$).

This code would be fine if we had the whole waveform that we wanted to filter in the **`filter_input`** array, and we were performing one-off operation – we could put the code in Listing 2 inside another loop that runs through the input data (stepping **`n`** in the outer loop) and writing the output to another array.

Things are not so simple for processing a continuous signal from the DMA buffer, which will be constantly changing as the DMA controller writes to it and will not contain the whole waveform. The mathematical operation is the same, but it takes a bit more work to get the correct data from the buffer to the calculation than using the static array implied by Listing 2.

## ADC DMA buffer operation

The filter code shares its basic structure with the ADC-to-DAC transfer operation in Listing 1. However, depending on the size of the buffer relative to the number of coefficients, we may need to access the whole buffer for one output sample calculation.

This is potentially a problem because only half the buffer can be regarded as stable at any time, with the other half being overwritten by the DMA controller. We also have to convert the integer data in the ADC buffer to floating-point values for the calculations. Both problems can be solved by copying the buffer contents to a separate floating-point array at the start of the two callback functions.

Listing 3 shows code snippets for the transfer and conversion of data from the ADC buffer (**`ADC_buffer`**) to the filter input buffer (**`filter_input`**). This code comprises loops placed at the start of the relevant callback function, like the ADC to DAC transfer in Listing 1, with the index range of the loop depending on the callback (which half of the buffer is being copied).

Similarly, the size of the ADC buffer (and filter input buffer) is defined as **`BUF_SIZE`**. The half-size of the buffer is also defined as **`HALF_BUF_SIZE`**.

The ADC buffer holds 12-bit integer values directly from the ADC via DMA, which are converted to floating point by the type-cast (**`(float)`**) in the copy operation. After the copy, operations on data in the whole filter input buffer can be used in filter calculations, unlike the ADC buffer, where only half the buffer can be used at one time.

## Data movement

To understand the code, it is helpful to track the movement and usage of data in the buffers in a small example – for the transfer and conversion just discussed, and later for the filter calculation. We will use a small buffer with 10 elements for the example (**`BUF_SIZE`** = 10, **`HALF_BUF_SIZE`** = 5). Usually, a real implementation would employ a larger buffer.

Fig.3 shows the state of the buffers after the first three samples $x[0]$, $x[1]$ and $x[2]$ have been read into the memory at the start of processing. In general, sample $x[n]$ is the latest sample and $x[n-1]$, $x[n-2]$ etc. are earlier samples. At this stage, the filter input buffer (**`filter_input`**) is empty.

When the buffer is half-full (as in Fig.4), the **`HAL_ADC_ConvHalfCpltCallback`** function is called. The code in Listing 3

Fig.3: an early ADC buffer snapshot.

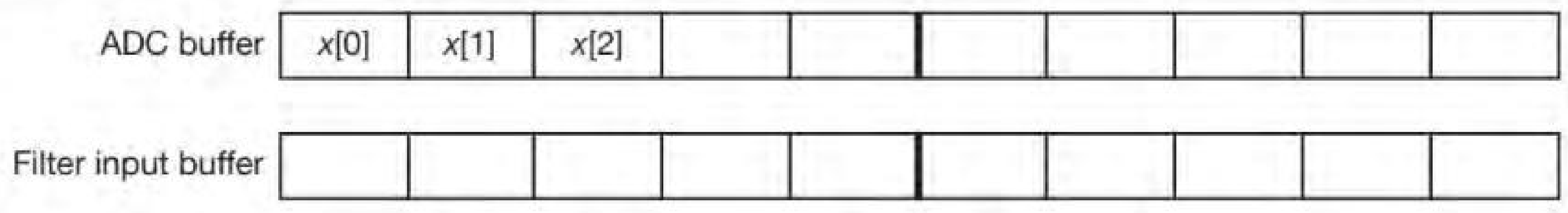
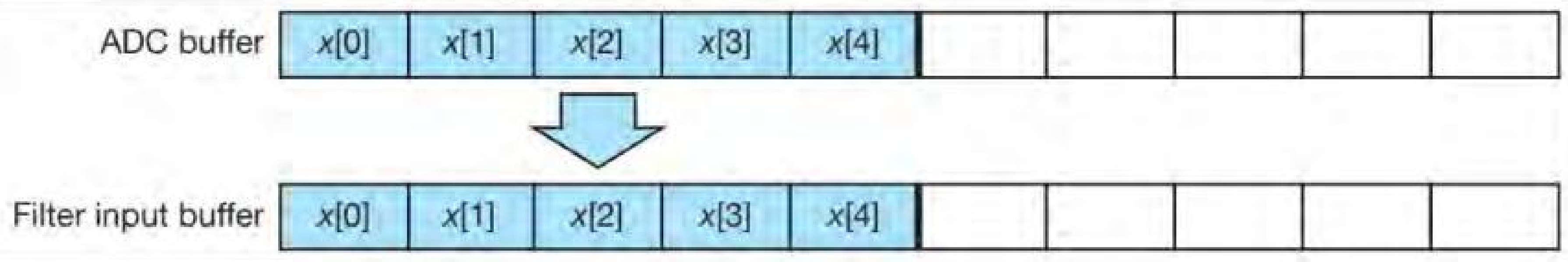


Fig.4: the first lower-half buffer transfer.



transfers data from the lower half of the ADC buffer to the filter input buffer, converting it from integer to floating point as it does so. The `HAL_ADC_ConvHalfCpltCallback` function also processes data to implement the filter, but this will be discussed later.

The DMA system continues to fill the ADC buffer. This happens in parallel with the data transfer, with the `HAL_ADC_ConvHalfCpltCallback` function also performing data processing, because the DMA system is separate from the processor running the code in the function. Fig.5 shows a further two samples in the ADC buffer.

When the buffer is full (see Fig.6), the `HAL_ADC_ConvCpltCallback` function is called. This code transfers the data from the upper half of ADC buffer to the filter input buffer, converting it from integer to floating point as it does so (and performing processing) – it performs the same function as the half full back, but on the other half of the buffer.

The DMA system writes values from the ADC to the ADC buffer continuously. When it gets to the end of the buffer, it returns to the start and overwrites the old data (circular buffer operation). Fig.7 shows the situation after the first wraparound – samples $x[10]$ and $x[11]$ have been written to the start of the buffer over the old data. The filter input buffer continues to hold the older data.

Every time the buffer is half-full, the `HAL_ADC_ConvHalfCpltCallback` function transfers a new batch of data from the lower half of the ADC buffer into the corresponding part of the filter input buffer (see Fig.8).

After the transfer, the filter input buffer contains the `BUF_SIZE` most recent samples (10 samples in this case). However, these are not in time-order through the whole buffer (as can be seen in Fig.8). The oldest samples are in the upper half of the buffer ($x[5]$ to $x[9]$ in the example), and the most recent samples are in the lower half ($x[10]$ to $x[14]$ in the example).

This makes processing a range of sample data within the function `HAL_ADC_ConvHalfCpltCallback` more complex than if they were sequential order.

Every time the buffer is full, the `HAL_ADC_ConvCpltCallback` function transfers a new batch of data from the upper half of the ADC buffer into the corresponding part of the filter input buffer (see Fig.9). After the transfer, the filter input buffer contains the `BUF_SIZE` most recent samples (10 samples in this case) in time order through the buffer.

The fact that the samples are in time order makes processing a sequence of samples simpler than in the case for half-full buffer.

Fig.5: more data arrives in the ADC buffer.

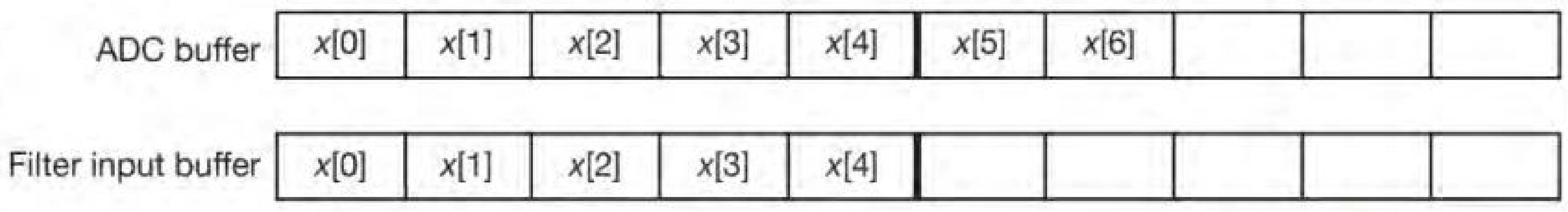

ADC buffer
x[0]
x[1]
x[2]
x[3]
x[4]
x[5]
x[6]
Filter input buffer
x[0]
x[1]
x[2]
x[3]
x[4]

Fig.6: the first upper half buffer transfer.

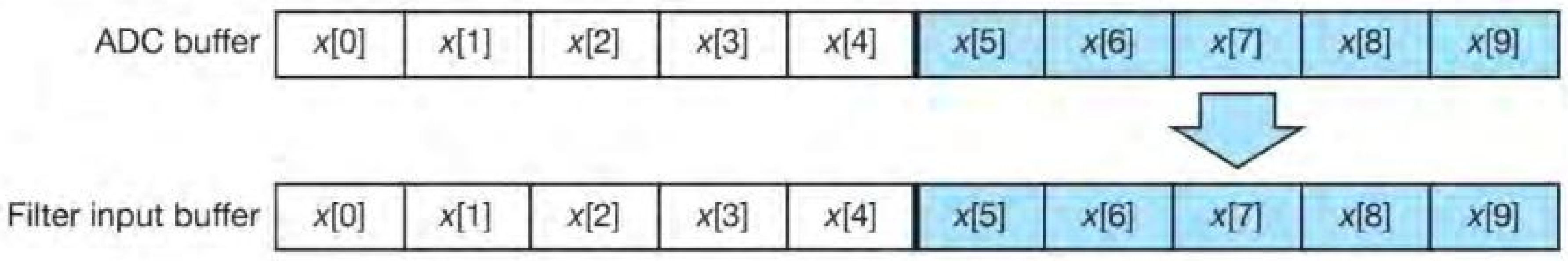

ADC buffer
x[0]
x[1]
x[2]
x[3]
x[4]
x[5]
x[6]
x[7]
x[8]
x[9]
Filter input buffer
x[0]
x[1]
x[2]
x[3]
x[4]
x[5]
x[6]
x[7]
x[8]
x[9]

Fig.7: the DMA unit wraps to the start.

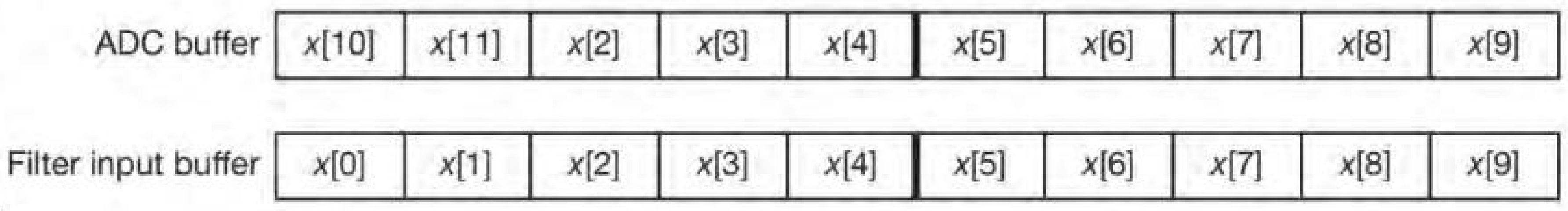

ADC buffer
x[10]
x[11]
x[2]
x[3]
x[4]
x[5]
x[6]
x[7]
x[8]
x[9]
Filter input buffer
x[0]
x[1]
x[2]
x[3]
x[4]
x[5]
x[6]
x[7]
x[8]
x[9]

Fig.8: the second half-full data transfer.

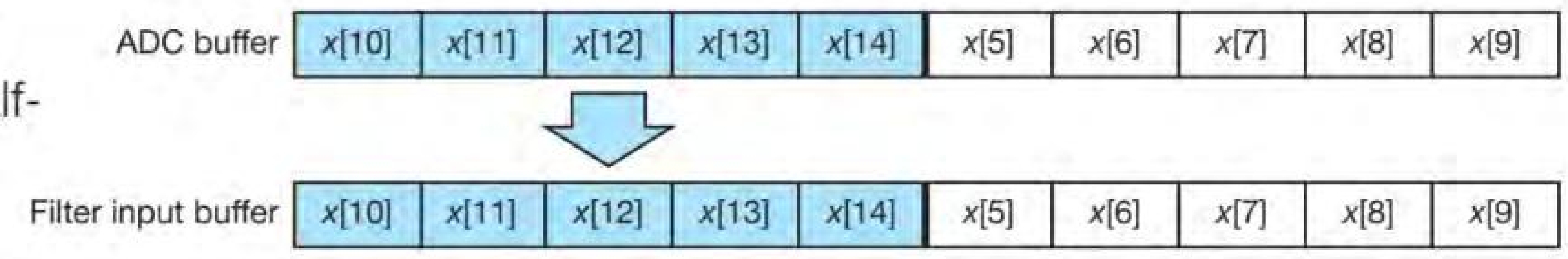

ADC buffer
x[10]
x[11]
x[12]
x[13]
x[14]
x[5]
x[6]
x[7]
x[8]
x[9]
Filter input buffer
x[10]
x[11]
x[12]
x[13]
x[14]
x[5]
x[6]
x[7]
x[8]
x[9]

Fig.9: the second buffer-full data transfer.

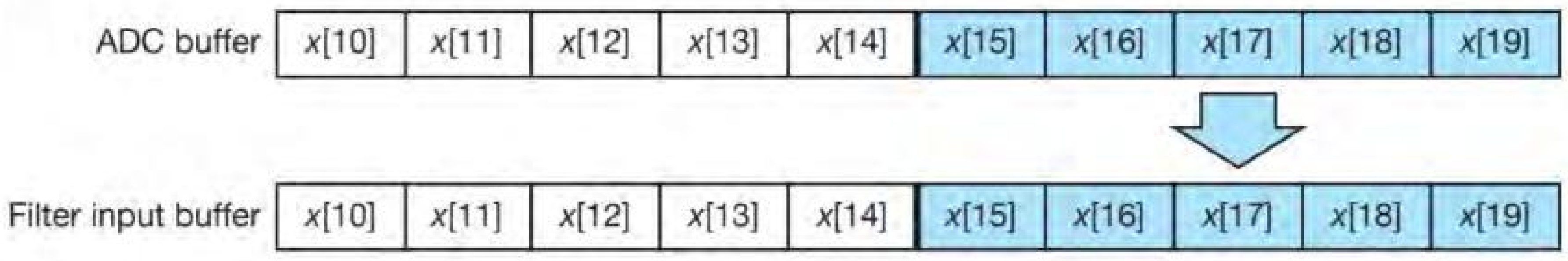

ADC buffer
x[10]
x[11]
x[12]
x[13]
x[14]
x[15]
x[16]
x[17]
x[18]
x[19]
Filter input buffer
x[10]
x[11]
x[12]
x[13]
x[14]
x[15]
x[16]
x[17]
x[18]
x[19]

Fig.10: how the output sample $y[n]$ is calculated from input samples $x[n-3]$ to $x[n]$.

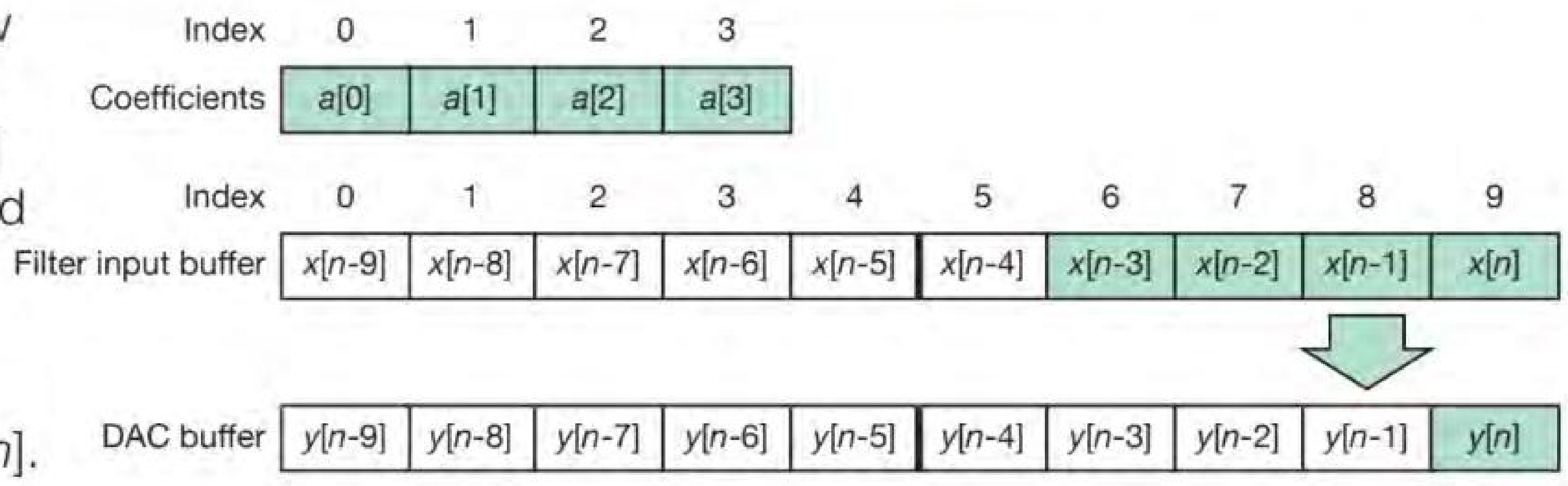
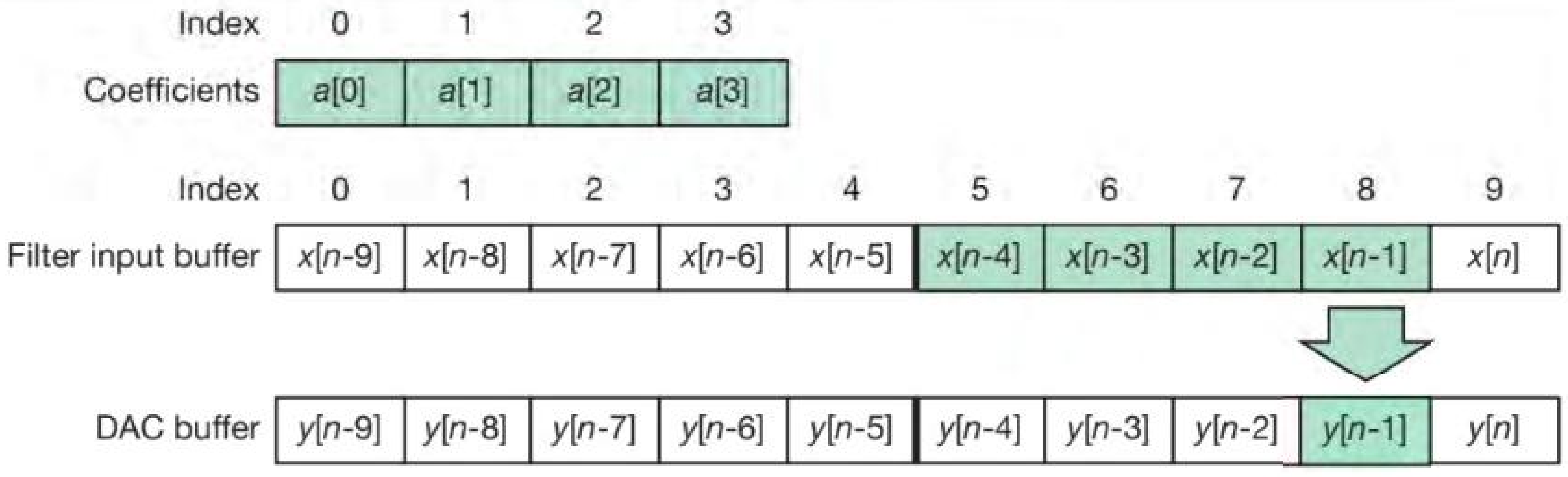

Index
0
1
2
3
Coefficients
a[0]
a[1]
a[2]
a[3]
Index
0
1
2
3
4
5
6
7
8
9
Filter input buffer
x[n-9]
x[n-8]
x[n-7]
x[n-6]
x[n-5]
x[n-4]
x[n-3]
x[n-2]
x[n-1]
x[n]
DAC buffer
y[n-9]
y[n-8]
y[n-7]
y[n-6]
y[n-5]
y[n-4]
y[n-3]
y[n-2]
y[n-1]
y[n]

Fig.11: output $y[n-1]$ is calculated from input samples $x[n-4]$ to $x[n-1]$.


Index
0
1
2
3
Coefficients
a[0]
a[1]
a[2]
a[3]
Index
0
1
2
3
4
5
6
7
8
9
Filter input buffer
x[n-9]
x[n-8]
x[n-7]
x[n-6]
x[n-5]
x[n-4]
x[n-3]
x[n-2]
x[n-1]
x[n]
DAC buffer
y[n-9]
y[n-8]
y[n-7]
y[n-6]
y[n-5]
y[n-4]
y[n-3]
y[n-2]
y[n-1]
y[n]

To perform the FIR filter calculations, it is necessary to access the correct data from the filter in buffer and coefficient arrays. In both callback functions, after data is transferred to the filter input buffer, it can be used, along with the coefficients array, to perform the FIR filter calculations. We will look at the buffer-full case first, because, as just noted, this is the more straightforward case.

## Buffer full callback data processing

After the transfer of data from the ADC buffer in the buffer-full callback, the filter input buffer contains the `BUF_SIZE` (10 in this case) most recent samples in order (see Fig.10). The oldest sample ($x[n-9]$ in this case) is in `FilterInBuffer[0]` (index 0) and the most sample recent in `FilterInBuffer[BUF_SIZE-1]` (index 9 in this case).

The FIR filter calculations for output $y[n]$ require the input values from sample $x[n]$ back to sample $x[n-N+1]$, where $N$ is the number of coefficients. For example, for four coefficients ($a[0]$, $a[1]$, $a[2]$ and $a[3]$) the calculation for $y[n]$ uses $x[n]$, $x[n-1]$, $x[n-2]$ and $x[n-3]$, as shown in Fig.10. The result is placed in the DAC buffer ready for the DMA system to send it to the DAC.

In the buffer-full callback function, it is necessary to calculate the values for all the output samples in the upper half of the DAC buffer ($y[n]$ down to $y[n-4]$ in this case). Fig.11 shows the values required to calculate $y[n-1]$.

This pattern is repeated down to the final value in the upper half of the DAC buffer ($y[n-4]$ in this case), which is calculated from $x[n-4]$ down to $x[n-7]$ – see Fig.12. Some calculations (such as that for $y[n-4]$) require data from the lower half of the buffer as well as the upper half.

Calculating the value for the bottom of the upper half of the DAC buffer uses mainly values from the lower half of the filter input buffer, specifically the top `COEFF_SIZE-1` values from that

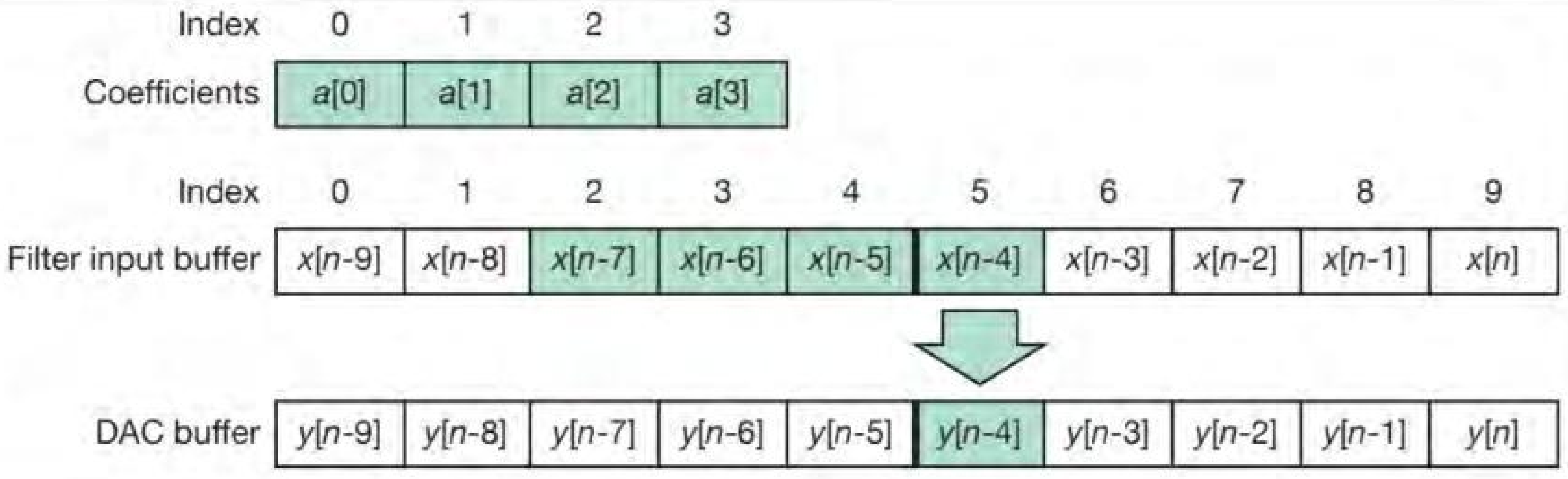

Index
0
1
2
3
Coefficients
a[0]
a[1]
a[2]
a[3]
Index
0
1
2
3
4
5
6
7
8
9
Filter input buffer
x[n-9]
x[n-8]
x[n-7]
x[n-6]
x[n-5]
x[n-4]
x[n-3]
x[n-2]
x[n-1]
x[n]
DAC buffer
y[n-9]
y[n-8]
y[n-7]
y[n-6]
y[n-5]
y[n-4]
y[n-3]
y[n-2]
y[n-1]
y[n]

Fig.12: output $y[n-4]$ is calculated from input samples $x[n-7]$ to $x[n-4]$.

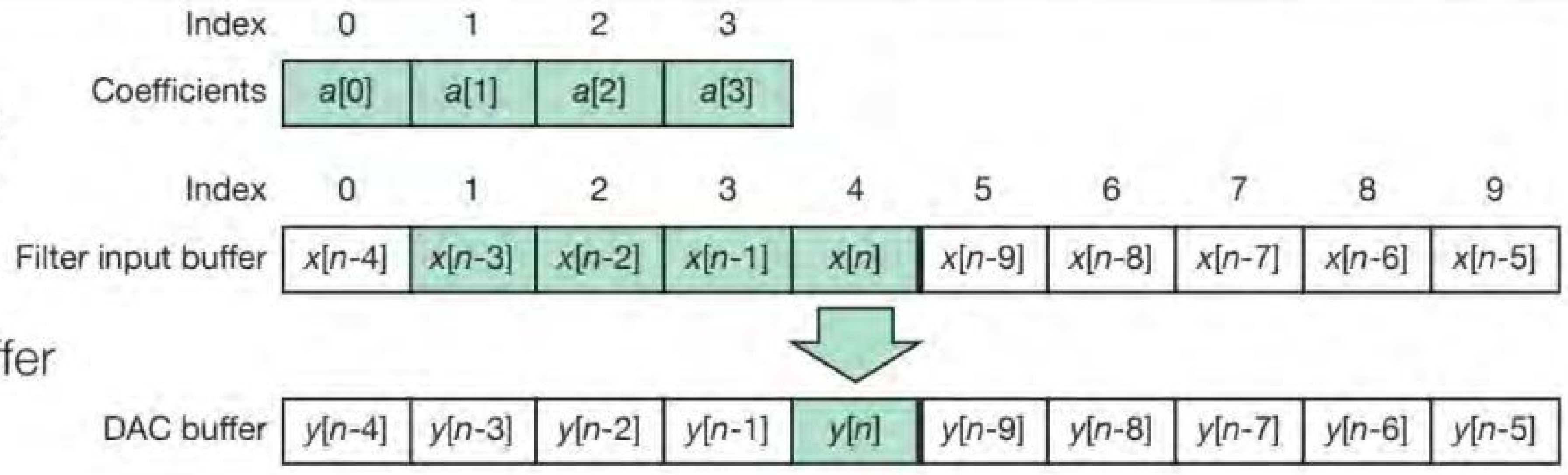

Index
0
1
2
3
Coefficients
a[0]
a[1]
a[2]
a[3]
Index
0
1
2
3
4
5
6
7
8
9
Filter input buffer
x[n-4]
x[n-3]
x[n-2]
x[n-1]
x[n]
x[n-9]
x[n-8]
x[n-7]
x[n-6]
x[n-5]
DAC buffer
y[n-4]
y[n-3]
y[n-2]
y[n-1]
y[n]
y[n-9]
y[n-8]
y[n-7]
y[n-6]
y[n-5]

Fig.13: calculating the output sample $y[n]$ in the half-full buffer callback.

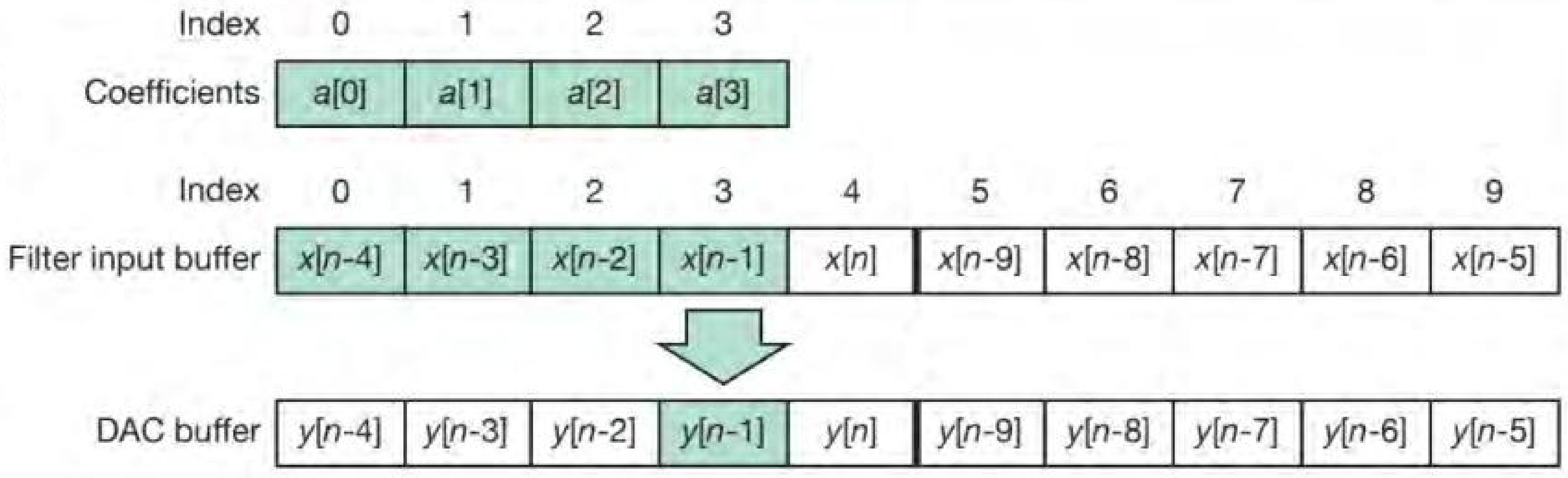

Index
0
1
2
3
Coefficients
a[0]
a[1]
a[2]
a[3]
Index
0
1
2
3
4
5
6
7
8
9
Filter input buffer
x[n-4]
x[n-3]
x[n-2]
x[n-1]
x[n]
x[n-9]
x[n-8]
x[n-7]
x[n-6]
x[n-5]
DAC buffer
y[n-4]
y[n-3]
y[n-2]
y[n-1]
y[n]
y[n-9]
y[n-8]
y[n-7]
y[n-6]
y[n-5]

Fig.14: calculating the output sample $y[n-1]$ in the half-full buffer callback.

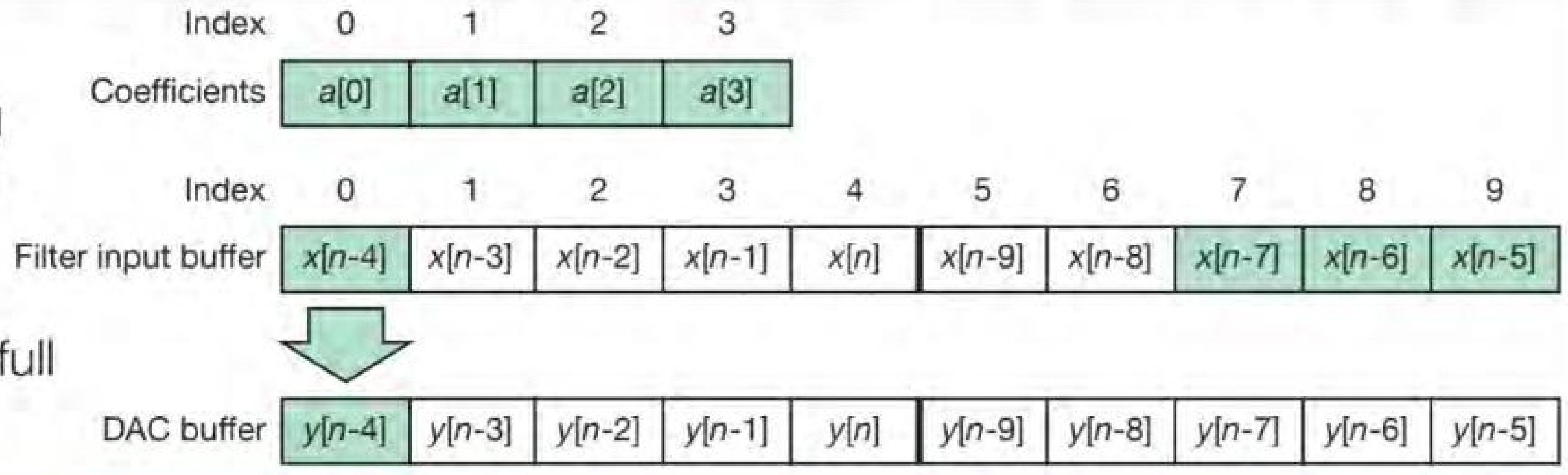

Index
0
1
2
3
Coefficients
a[0]
a[1]
a[2]
a[3]
Index
0
1
2
3
4
5
6
7
8
9
Filter input buffer
x[n-4]
x[n-3]
x[n-2]
x[n-1]
x[n]
x[n-9]
x[n-8]
x[n-7]
x[n-6]
x[n-5]
DAC buffer
y[n-4]
y[n-3]
y[n-2]
y[n-1]
y[n]
y[n-9]
y[n-8]
y[n-7]
y[n-6]
y[n-5]

Fig.15: calculating the output sample $y[n-4]$ in the buffer-full callback.

```
1   for (int n = HALF_BUF_SIZE; n < BUF_SIZE; n++)
2   {
3     Result = 0;
4     for (int i = 0; i < COEFF_SIZE; i++)
5     {
6       Result = Result + Coefficient[i] * filter_input[n-i];
7     }
8     DAC_buffer[n] = (uint32_t)Result;
9   }
```

Listing 4: the filter calculations for a buffer-full callback.

half. This means that the maximum number of coefficients we can have is **HALF_BUF_SIZE+1**. Otherwise, we would run off the bottom of the filter input buffer.

In all cases for the full buffer callback, the input samples (*x* values) are in a continuous block in the filter input buffer. This makes it easy to iterate over the input values required to calculate a given output using a single for loop. The data in the lower half of the filter input buffer is stable and safe to use as long as the full callback function completes running before the next half-full callback.

## FIR filter code for buffer-full callback

The buffer-full callback must calculate all the output values in the upper half of the DAC buffer (indexes from **HALF_BUF_SIZE** to **BUF_SIZE-1**). This can be achieved with the code from Listing 2, with **n** running over the required index range (see Listing 4).

After the calculation loop, each value of **Result** (*y* values) is converted to an integer and stored in the DAC buffer for output to the DAC via DMA.

For the example buffers shown above (**BUF_SIZE** = 10 and **COEFF_SIZE** = 4), the code in Listing 4 performs the following FIR filter calculations, where *a*[*i*] is the value at index *i* in the coefficient array and *b*[*i*] is the value at index *i* in the filter input buffer array.

$$\begin{aligned}
y[n-4] &= a[0]b[5] + a[1]b[4] + a[2]b[3] + a[3]b[2]\\
y[n-3] &= a[0]b[6] + a[1]b[5] + a[2]b[4] + a[3]b[3]\\
y[n-2] &= a[0]b[7] + a[1]b[6] + a[2]b[5] + a[3]b[4]\\
y[n-1] &= a[0]b[8] + a[1]b[7] + a[2]b[6] + a[3]b[5]\\
y[n] &= a[0]b[9] + a[1]b[8] + a[2]b[7] + a[3]b[6]
\end{aligned}$$

## Buffer half-full callback FIR calculation

Like the buffer-full callback, after the transfer of data from the ADC buffer in the half-full callback, the filter input buffer contains the **BUF_SIZE** most recent samples. However, the sample order is more complex than in the buffer-full case. Fig.13 shows the situation at a buffer half-full callback. The filter input buffer is in a state equivalent to that shown in Fig.8, with *n* = 14.

The newest samples are in the first half of the filter input buffer. The first half ends with the newest input sample, *x*[*n*], in **filter_input[HALF_BUF_SIZE -1]** (index 4 in this case).

The oldest samples are in the second half of the buffer; the second half starts with the oldest sample being at **filter_input[HALF_BUF_SIZE]** (index 5 in this case) and runs in sequence to the final sample index at **filter_input[BUF_SIZE-1]** (ie, **filter_input[9]**). The next newest sample is at index 0, at the bottom of the first half.

The FIR filter calculations require the same data from the filter input buffer as the buffer-full callback, but the location of the data in the filter input buffer is different. Calculation for *y*[*n*] uses *x*[*n*], *x*[*n* – 1], *x*[*n* – 2] and *x*[*n* – 3], and the coefficients, as shown in Fig.13. This is similar to the example in Fig.10, but using lower-half data.

In the buffer half-full callback, it is necessary to calculate the values for all the output samples in the lower half of the DAC buffer (*x*[*n*] down to *x*[*n* – 4] in this case). The example in Fig.14 shows the values used to calculate *y*[*n* – 1].

This pattern is repeated down to the final value in the lower half of the DAC buffer. For example, *y*[*n* – 4] is calculated from *x*[*n* – 4] down to *x*[*n* – 7] and the coefficients (see Fig.15). Some calculations (such as that for *y*[*n* – 4]) require data from the upper half of the filter input buffer as well as the lower half.

The data values used are the same as for the buffer full callback, but finding the correct index to access the data in the buffer is more complex because the data is in two separate blocks in the lower and upper halves.

In a similar way to the buffer full case, data in the upper half of the filter input buffer is stable and safe to use as long as the half-full callback function completes running before the next full callback. The maximum number of coefficients is the same for this case.

## FIR filter code for buffer half-full callback

The mathematics for the FIR filter calculation in the half-full callback function is the same as for the full callback function, so the FIR filter calculation is the same as the full buffer case (line 6 in Listing 4). However, the code is more complex because the buffer indices of the data do not always form a continuous block. If the data is in two groups (as in the $y[n-4]$ example above), two loops are required.

The input buffer index is $n - i$ for a calculation of $y[n]$ where $i$ is the coefficient index. Therefore, if $n - i$ is greater than zero for the largest value of $i$ (the last value in the coefficient loop), the set of input sample values ($x$) used in the calculation will all be in a continuous block in the lower half of the buffer, so the same loop code as the buffer-full case can be used (Listing 4, lines 3 to 7).

The largest value of $i$ is $N - 1$ (for $N$ coefficients), so the lowest index in a potentially continuous block of sample data is $n - N + 1$. If this index is zero or above, then a simple loop can be used. The code in Listing 5 finds the lowest index (the integer `Lowest_index` in the code, see line 2), and if this is greater than or equal to zero, runs the calculation for $y[n]$ in the same way as the buffer-full callback case.

If the value of `Lowest_index` calculated above is less than zero, some of the input samples will be in the upper half of the buffer. Two loops need to be used, one to iterate over the values in the lower half and the other for the upper half (this is shown in Listing 6, which is the "else" part of the "if" in Listing 5).

The first loop needs to take the value of the coefficient index (loop counter `i`) from 0 to `n` (for $y[n]$), so the input sample index $n - i$ ranges down to zero (see Listing 6, lines 3 to 6). The value of `i` reached by this first loop is used to determine the range of the second loop.

The number of samples covered by the second loop is equal to the number of coefficients remaining after the first loop. As $n$ coefficients were covered in the first loop, this number is $N - n$. In the code, the integer `Remaining_Samples` is used to hold this value and hence control iterations of the second loop (see Listing 6, line 7).

The filter input sample values required by the second loop start at the top of the filter input buffer (at index `BUFF_SIZE` – 1) and range down to `BUFF_SIZE - Remaining_Samples`. Thus, the loop index (`i`) needs to run from 1 to `Remaining_Samples`, and the filter input buffer can be indexed using `BUFF_SIZE - i` (see Listing 7, lines 8 to 11).

At the same time, the coefficient index has to range from $n + 1$ (the last coefficient index in the first loop was $n$) to $N - 1$. In the code, we use a separate integer in the loop (c), initialised to $n + 1$, and incremented on each iteration, to index the coefficients array.

After the calculation loop(s) (whichever is used), each value of `Result` (output value $y$) is converted to an integer and stored in the DAC buffer for output to the DAC via DMA. This is the same as the buffer-full callback (Listing 4, line 8).

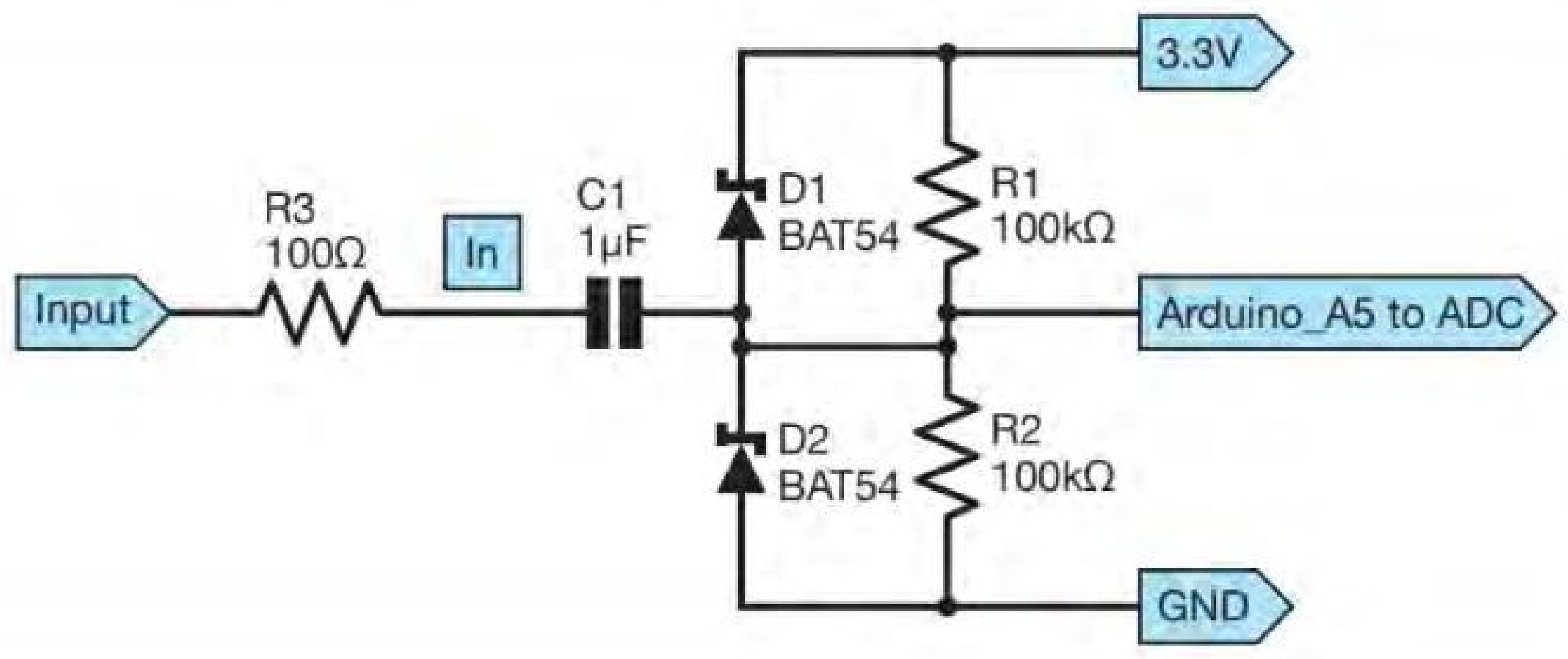
```
1  Result = 0;
2  Lowest_index = n - COEFF_SIZE + 1;
3  if (Lowest_index >= 0)
4  {
5    for (int i = 0; i < COEFF_SIZE; i++)
6    {
7      Result = Result + Coefficient[i] * filter_input[n-i];
8    }
9  }
```

Listing 5: the single loop used when all necessary samples are in a continuous block.

```
1  for (int i = 0; i <= n; i++)
2  {
3    Result = Result + Coefficient[i] * filter_input[n-i];
4  }
5  Remaining_Samples = COEFF_SIZE - n;
6  for (int i = 1, c = n+1; i <= Remaining_Samples; i++, c++)
7  {
8    Result = Result + Coefficient[c] * filter_input[BUF_SIZE - i];
9  }
```

Listing 6: two loops used when the required samples are not continuous.

Fig.16: the input protection circuit I added to the microcontroller board for testing the filter implementation.



For the example buffers shown above, the code performs the following FIR filter calculations; the notation is as in the previous example:

$$\begin{aligned}
y[n-4] &= a[0]x[0] + a[1]x[9] + a[2]x[8] + a[3]x[7] \\
y[n-3] &= a[0]x[1] + a[1]x[0] + a[2]x[9] + a[3]x[8] \\
y[n-2] &= a[0]x[2] + a[1]x[1] + a[2]x[0] + a[3]x[9] \\
y[n-1] &= a[0]x[3] + a[1]x[2] + a[2]x[1] + a[3]x[0] \\
y[n] &= a[0]x[4] + a[1]x[3] + a[2]x[2] + a[3]x[1]
\end{aligned}$$

The first three of these require the two loops; the last two use the single loop.

The majority of code for the filtering operation is shown in Listing 7. This includes the #defines of the buffer and coefficient array sizes (lines 34 to 36); the coefficient array initialisation (lines 68 to 79); declaration of the variables used in the filter calculations (lines 81 to 86); and the callback functions (lines 872 to 934), which include the snippets in Listings 3 to 6 discussed above.

The callbacks also control a timing pulse, as in previous examples. Listing 7 does not include the DMA and timer initialising in the User Code 2 section, as this is the same as the example last month.

## Implementation details

The filter was implemented using a new STM32CubeIDE project configured in the same way as the ADC-to-DAC DMA example from last month. An example filter was configured with a 1kHz low-pass response using 37 coefficients and sampling at 48kHz. The filter coefficients were obtained using the online tool at fiiir.com (this site was introduced in the May issue).

Last month, we discussed the ADC input protection and DC shift circuit with reference to LTspice simulations that included parasitic capacitance and a model of the ADC input. Fig.16 shows the circuit I constructed, rather than the model. I built it on an Arduino Uno Prototyping Shield (TSX00083).

The B-L475E-IOT01A development board for the microcontroller provides Arduino Uno V3 connectivity, so can conveniently be used with Arduino shield boards.

The Prototyping Shield has an array of through-hole solder pads for mounting components. Unlike stripboard, these are not connected in rows, so the component leads and interconnect wires are soldered together (after suitable bending and cutting) on the underside of the board.

The input to the protection circuit is provided by a 3-way pin header (the third pin provides for monitoring/measuring the signal, if required). A 2-way pin header was connected to the row of solder pads connected to ground

Listing 7: the full code for the FIR filter parts of the project.

```
/* USER CODE BEGIN PD */
#define BUF_SIZE 76
#define HALF_BUF_SIZE 38
#define COEFF_SIZE 37 // Must be smaller than HALF_BUF_SIZE+2
/* USER CODE END PD */
…
/* USER CODE BEGIN PV */
// FIR filter coefficients, fc =1 kHz, fs = 48 kHz, rectangular window
const float Coefficient[COEFF_SIZE] = {
0.011167887305860052, 0.013267107833265871, 0.015387539289897745, 0.017509857539699709,
…
0.011167887305860052
};

float Result;                    // Fir Filter output value
int Lowest_index;                // Used for buffer index tracking in DMA callback
int Remaining_Samples;           // Used for buffer index tracking in DMA callback
uint32_t ADC_buffer[BUF_SIZE]; // DMA ADC input buffer
uint32_t DAC_buffer[BUF_SIZE]; // DMA DAC output buffer
float filter_input[BUF_SIZE];  // input buffer for FIR calculations
uint32_t Prescaler = 1;          // Timer prescaler setting
float Period;                    // Timer period in us
float Frequency = 48.0;          // Sample frequency in kHz
/* USER CODE END PV */
…
/* USER CODE BEGIN 4 */
void HAL_ADC_ConvHalfCpltCallback(ADC_HandleTypeDef* hadc)
{
  HAL_GPIO_WritePin(GPIOB, ARD_D8_Pin, GPIO_PIN_SET); // Timing monitoring pulse on
  // Copy DMA data to filter input buffer
  for (int i = 0; i < HALF_BUF_SIZE; i++)
  {
    filter_input[i] = (float) ADC_buffer[i];
  }
  // Calculate FIR
  for (int n = 0; n < HALF_BUF_SIZE; n++)
  {
    Result = 0;
    Lowest_index = n - COEFF_SIZE + 1;
    if (Lowest_index >= 0)
    {
      // All samples in contiguous block in lower half of buffer - simple loop
      for (int i = 0; i < COEFF_SIZE; i++)
      {
        Result = Result + Coefficient[i] * filter_input[n-i];
      }
    } else {
      // Samples not contiguous - need two loops
      // First loop in lower half of buffer (index down to zero)
      for (int i = 0; i <= n; i++)
      {
        Result = Result + Coefficient[i] * filter_input[n-i];
      }
      // Second loop in upper half of buffer (working back from the top)
      Remaining_Samples = COEFF_SIZE - n;  // Total samples minus number already done
      for (int i = 1, c = n + 1; i <= Remaining_Samples; i++, c++)
      {
        Result = Result + Coefficient[c] * filter_input[BUF_SIZE - i];
      }
    }
    // Copy result to DAC DMA buffer
    DAC_buffer[n] = (uint32_t)Result;
  }
  HAL_GPIO_WritePin(GPIOB, ARD_D8_Pin, GPIO_PIN_RESET); // Timing monitoring pulse off
}

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
  HAL_GPIO_WritePin(GPIOB, ARD_D8_Pin, GPIO_PIN_SET); // Timing monitoring pulse on
  // Copy DMA data to filter input buffer
  for (int i= HALF_BUF_SIZE; i < BUF_SIZE; i++)
  {
    filter_input[i] = (float) ADC_buffer[i];
  }
  // Calculate FIR
  for (int n= HALF_BUF_SIZE; n < BUF_SIZE; n++)
  {
    Result = 0;
    for (int i = 0; i < COEFF_SIZE; i++)
    {
      Result = Result + Coefficient[i] * filter_input[n-i];
    }
    // Copy result to DAC DMA buffer
    DAC_buffer[n] = (uint32_t)Result;
  }
  HAL_GPIO_WritePin(GPIOB, ARD_D8_Pin, GPIO_PIN_RESET); // Timing monitoring pulse off
}
/* USER CODE END 4 */
```

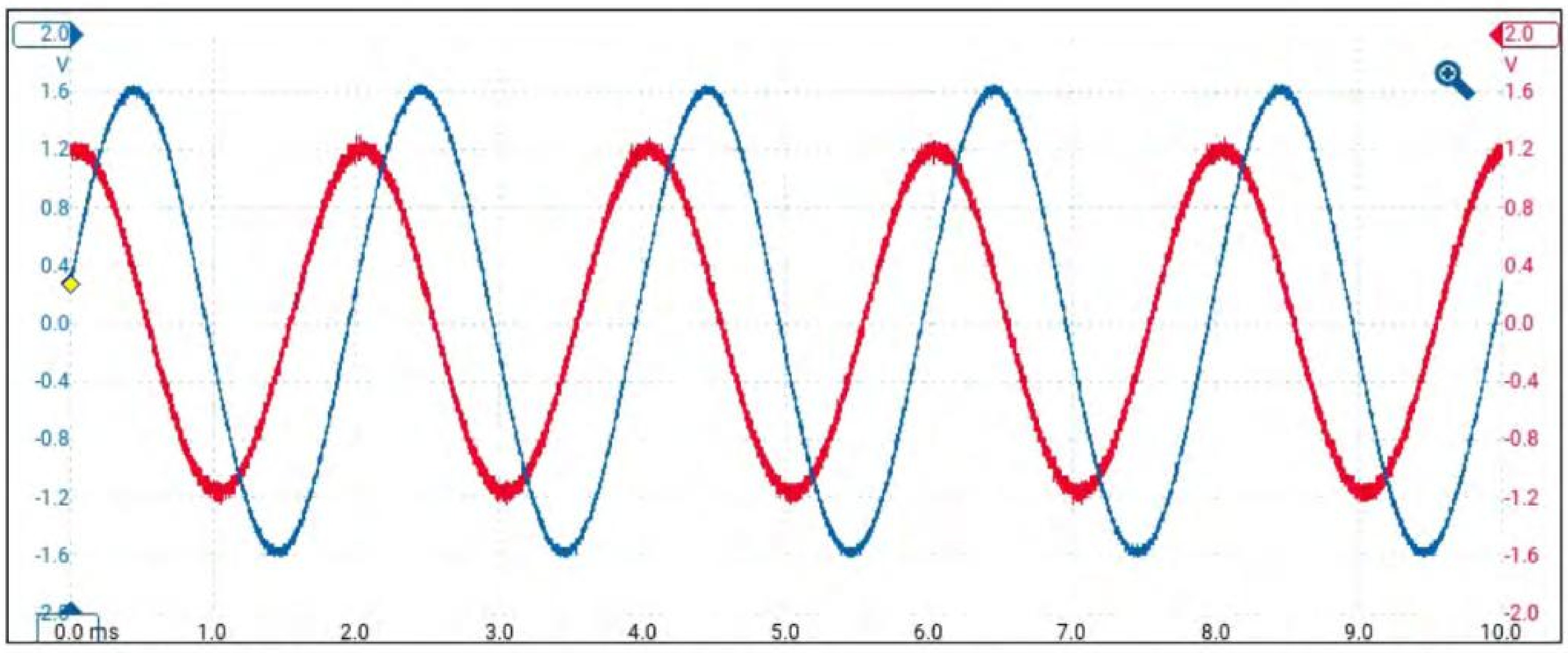


Fig.18: the 500Hz, 3.15V peak-to-peak input signal (blue) and the output signal (red).

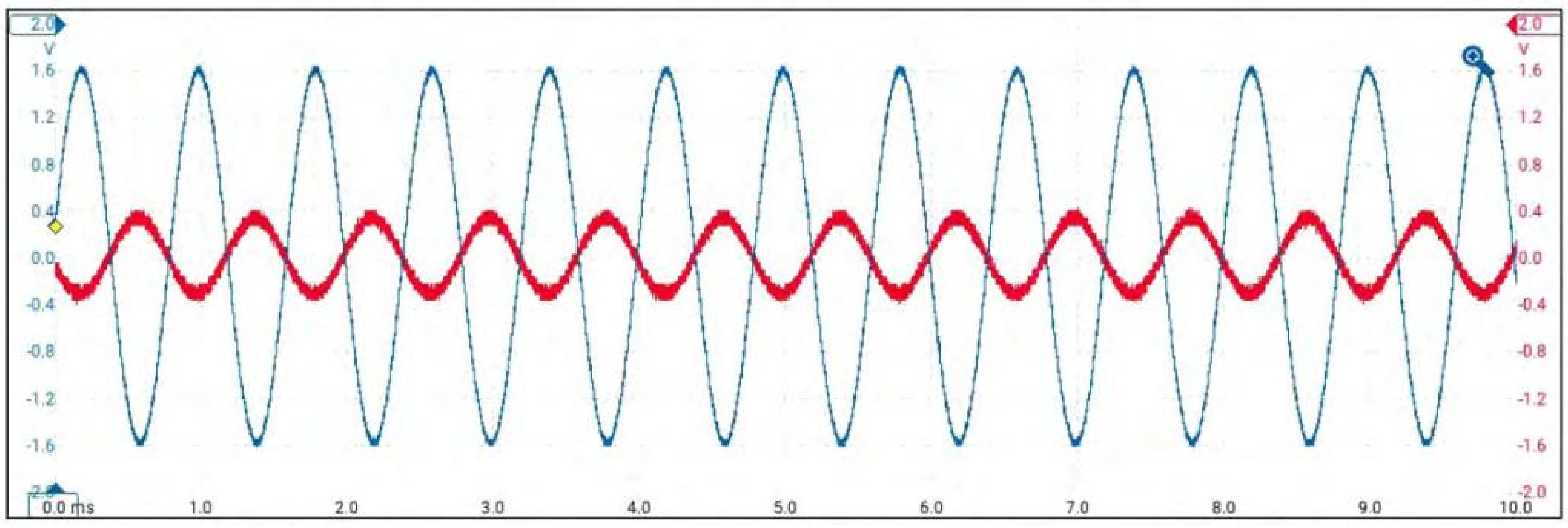


Fig.19: the 1250Hz, 3.15V peak-to-peak input signal (blue) and output signal (red).

(visible on the right edge of the Prototyping Shield in Fig.17). This is useful for connecting test instrument ground clips.

The BAT54 schottky diodes used in the protection circuit are available in a few formats, including single and dual-diode configurations in different SMD packages (eg, SOT-23, SC-70 and SOD-323). I chose the single diode BAT54WS-G variant as it is the best fit to the 1.27mm (0.05in/50 thou) pitch SMD solder pads on the Prototyping Shield (these are intended for an SOIC integrated circuit).

As inputs containing frequencies above the Nyquist rate were not going to be used for testing, no anti-aliasing filter was provided at the input. A simple first-order RC low-pass reconstruction filter with $R = 8.2k\Omega$ and $C = 8.2nF$ was used on the DAC output. This has a cutoff frequency of 2.4kHz ($1 \div 2\pi RC$).

At the Nyquist rate of 24kHz for the example digital filter, the RC filter has an attenuation of about 20dB (⅒th), which is sufficient for testing with sinewave inputs and a filter cutoff of 1kHz.

The filter was tested by applying sinewaves at close to the maximum amplitude (3.15V peak-to-peak, just within the ADC range of 3.3V). This confirmed that the gain dropped at frequencies above 1kHz. Two examples are shown in Figs.18 & 19, which are for 500Hz and 1250Hz, respectively.

Ideally, we need to plot the frequency response. This could be done manually by measuring the gain a various frequencies using measurements, such as in Figs.18 & 19, but this would be a tedious process. Next month, we will look at automated frequency response measurement. **PE**

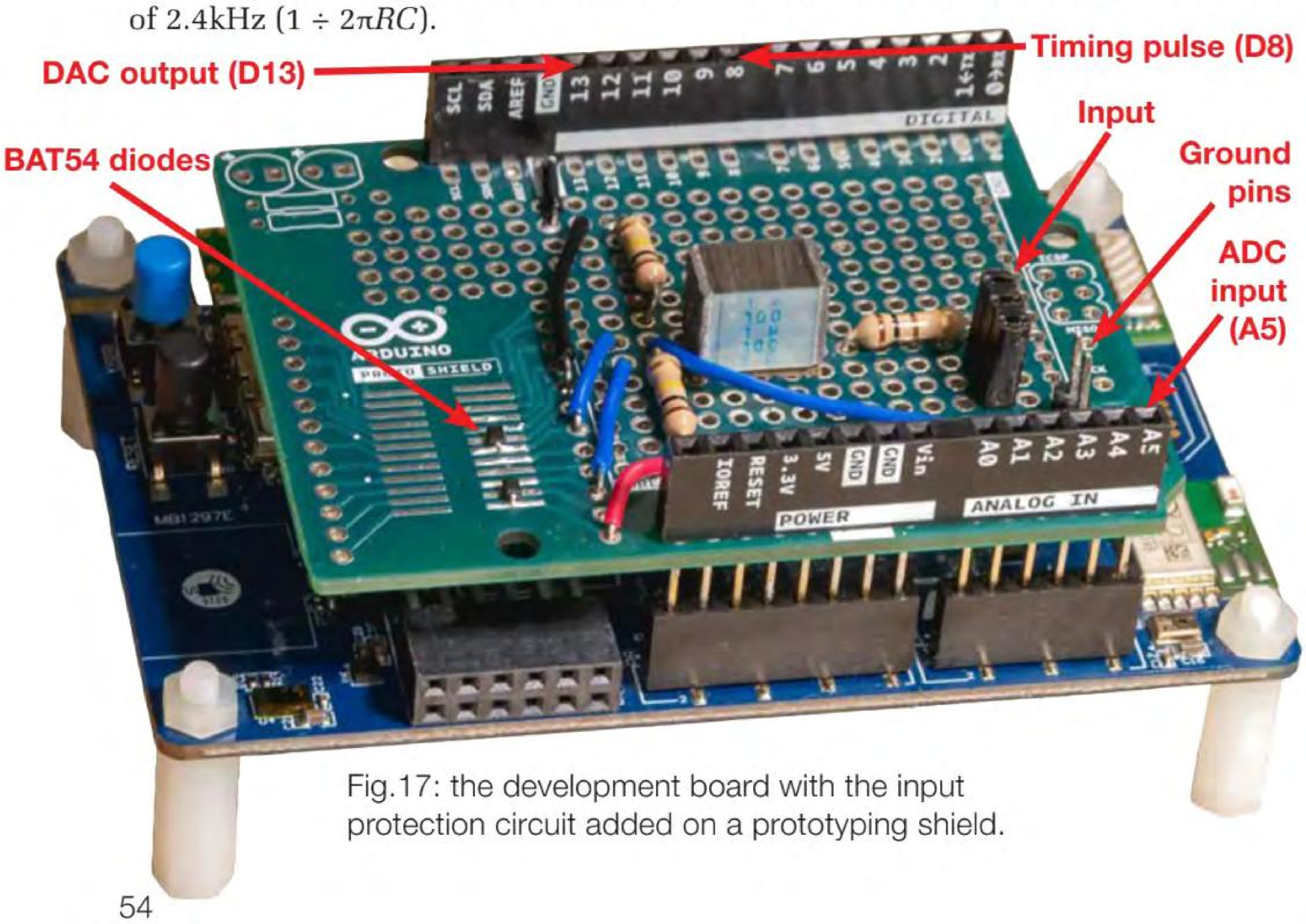


Fig.17: the development board with the input protection circuit added on a prototyping shield.

PoLabs
For more information or software download please visit
www.poscope.com/epe
PoKeys
Connect, Control
- USB
- Ethernet
- Web server
- Modbus
- CNC (Mach3/4)
- IO
- PWM
- Encoders
- LCD
- Analog inputs
- Compact PLC
Stepper motor drivers
Drive
- up to 256 microsteps
- 50 V / 6 A
- USB configuration
- Isolated
- up to 32 microsteps
- 30 V / 2.5 A
PoScope Mega1+
PoScope Mega50
Measure
- up to 50MS/s
- resolution up to 12bit
- Lowest power consumption
- Smallest and lightest
- 7 in 1: Oscilloscope, FFT, X/Y, Recorder, Logic Analyzer, Protocol decoder, Signal generator

# AOShop

The home for specialist audio, analogue and historic components – provided by Jake Rothman, *PE's Audio Out* columnist. The AOShop is your best bet for classic analogue 'NOS' (new old stock) components, including all parts for *Audio Out* projects and designs.

## Ge semiconductors

| | |
|---|---|
| *Small-signal PNP transistors* | |
| NKT214F, OC57, OC59, 2N1377, 2N525 | £1.00 |
| *Low-noise PNP transistors* | |
| GET106 | £2.50 |
| *Small power PNP transistors* | |
| AC153, AC153K, AC188 | £1.50 |
| *Small power NPN transistors* | |
| AC176, AC141K | £2.00 |
| *High-voltage PNP transistors* | |
| OC77, CV7001 | £1.50 |
| *TO3 PNP power transistors* | |
| OC22, CV7054 (OC23), OC25, OC35, OC36, AD143, AD149, AD161, AD162 | £2.00 |
| AD140 | £3.50 |
| AD149, AD161/2 matched pair | £5.00 |
| *RF PNP transistors* | |
| OC41, OC42, 2SA12, 2SA53,AF124, AF178, GET872A | £1.50 |
| *NPN transistors* | |
| OC139, OC140, ASY73 | £2.00 |
| *Diodes* | |
| CV7049 (OA10), CG92 (OA91) | £0.50 |

## Si semiconductors

| | |
|---|---|
| *Diodes* | |
| ZC5800 RF Schottky | £0.20 |
| *Low-noise silicon transistors* | |
| BFW16A, 2SC3071, 2SC3068, 2SA1016K, 2SC2362K, 2SA970BL | £1.50 |
| 2SC2204, 2SD655, BC550C | £0.50 |
| ZTX651 | £0.30 |
| ZTX751 | £0.50 |
| *RF transistors (suitable for Theremin)* | |
| BF199 | £0.50 |
| *Audio power MOSFETs* | |
| Exicon 10N20, 10P20 | £6.50 |
| Hitachi 2SJ99, 2SK343 | £3.50 |
| Hitachi 2SJ56, 2SK176 | £8.50 |
| *JFETs* | |
| BFW11, BFW10, TIS73L, J177, J113, U1994, U1898, 2SJ176, J201 | £1.00 |
| J175, J176, J112, J111, 2N3820, 2N5467, BF244, 2N5460, J230 | £0.60 |
| *Small power output/driver transistors* | |
| 2SB649A, D669A, 2SA1208, 2SC2910, MJE253G, MJE243G, 2SA1725, 2SC4511 | £1.20 |
| BD139,BD140, BD135, BD156, BD435, BD436, MPSA63, BCV46 | £0.50 |
| *MOSFETs* | |
| ZVP2106A | £0.30 |
| Dual-gate 3SK45, BFS28 | £1.50 |
| *Metal-cased transistors* | |
| BCY71 | £0.30 |
| *Amplifiers* | |
| LM318 high-speed op amp | £0.35 |
| µA709 metal-cased op amp | £2.00 |
| 5534H metal-cased op amp | £3.00 |
| LM384, TDA2030A, TDA2050V | £1.50 |
| TAA435 (Mullard power amp driver for AD161/AD162) | £1.20 |

## Synthesiser ICs

| | |
|---|---|
| That/dbx 2180 VCA/VCF | £6.00 |
| CA3080 VCA/VCF (vocoder) | £2.80 |
| CA3280 VCA/VCF | £4.00 |
| LM13600/ LM13700 VCA/VCF | £2.00 |
| CA3086 transistor array | £1.00 |
| PT2399 echo/delay | £1.50 |

## Dual transistors

| | |
|---|---|
| 2N2639, 2N2223, 2N2910 (NPN) | £4.00 |
| E401 (JFET Moog) | £4.00 |
| 2SK2145-Y dual JFET | £0.80 |
| 2N5564 JFET | £8.00 |
| HN3C51F, HN3A51F | £1.00 |
| DMMT3904/6, HN1A01F, HN1C01F | £0.50 |

## Loudspeakers

| | |
|---|---|
| *PE Mini-Monitor* | |
| Volt PE165 6.5-inch woofer (each) | £85 |
| Morel MDT29 tweeter (each) | £25 |
| Kit *pair* of PE165/MDT29 plus Volt crossover parts and PCBs | £299 |
| Monacor DT-28N tweeter (each) | £35 |
| Vifa 19mm BC20SC15-04 tweeter (each) | £15 |
| *Volt crossover inductors* | |
| 1.2mH, 1.5mH, 2mH, 2.7mH, 0.5mH (tapped at 0.3mH) (each) | £5.85 |
| *Fully assembled and tested high-quality speaker prototypes – ask for details* | |
| LS3/5As and other similar speaker systems (pair) | £200-£350 |
| *Fully tested reclaimed speakers* | |
| Vifa BC14 5-inch woofer (each) | £10 |
| Vifa TC26 1-inch tweeter (each) | £10 |
| *Low-price speakers* | |
| Philips 4-inch 4070 | £2.00 |
| EMI 10x6-inch, 30Ω Alnico | £7 |
| 64mm 64Ω neodymium | £1.20 |

## Capacitors

***Note '10/63' denotes '10µF 63V'.***

| | |
|---|---|
| *Polyester* | |
| 3.3/100, 4.7/250, 4.7/63 | £1.00 |
| 5.6/63, 8.2/63, 10/63 | £2.00 |
| Mullard 'Mustard' C296 0.22/400 | £2.00 |
| *Polycarbonate* | |
| Axial 2.2/63 1%, 4.7/160, 6.8/63 | £1.00 |
| *Radial* | |
| 6.8/160V, 10/63 | £2.00 |
| Reclaimed 22/63 | £2.00 |
| *Polystyrene* | |
| Philips 1% 4.7nF/160, 6.2nF/500, 12nF/63, 22nF/63, 110nF/63, 24nF, 2nF | £1.00 |
| RIFA 1% 100nF/100, Suflex 90.9nF 0.5% | £2.00 |
| Suflex 2.5% 10nF/63 (rad. or ax.) | £0.50 |
| *Radiation resistant* | |
| Siemens cellulose acetate MKL 2.2/25 | £0.80 |
| *Electrolytic – Mullard blue 017 series* | |
| 10/25, 22/25, 100/10 | £0.50 |
| 150/40, 470/40, 1000/40 | £1.00 |
| *Tantalum – axial metal cased* | |
| 22/50, 47/35, 68/25, 100/20, 120/10, 150/16, 220/10, 330/6 | £2.00 |
| 22/35, 33/35, 47/20, 68/15, 100/10, 150/6 | £1.25 |
| 4.7/50, 6.8/35, 10/25, 10/35, 22/15 | £1.00 |
| *Axial moulded-case tantalum* | |
| Kemet axial 6.8/10 | £0.30 |
| Kemet radial 33/10 | £0.40 |
| STC radial 100/20 | £1.50 |
| *Tantalum bead* | |
| 22/50 | £1.00 |
| 470/3 | £2.00 |
| 680/6.3 | £3.00 |
| *Wet tantalum* | |
| 220/25 axial | £3.20 |
| Castanet button 140/30, 470/3 | £3.20 |
| Hughes 540/10 | £3.20 |
| *Bipolar* | |
| Hermetic bipolar tantalum 16/35 | £3.20 |
| Elcap axial 10/50 | £0.50 |
| Generic radial 100/16, 470/35, 100/ 63, 22/35, 4.7/35, 220/16 | £0.50 |
| *Philips solid-aluminium (axial) 121/123* | |
| 47/16 | £1.00 |
| 330/6.3 | £2.00 |
| 100/35 | £4.00 |
| *Philips Pearl 122 series (radial)* | |
| 10/16 | £0.30 |
| *Silvered mica (radial)* | |
| 1nF/500 1% | £1.00 |
| *Trimmer capacitors* | |
| Vishay plastic-film 4-40pF | £0.80 |
| Vishay plastic-film 5.5-45pF | £1.00 |
| Vishay plastic-film 5-80pF | £1.20 |
| Vishay plastic-film 10-250pF | £2.00 |
| Mica 1-12pF, 2-40pF | £1.00 |

## Audio transformers and inductors

| | |
|---|---|
| *Eagle transformers* | |
| LT44, LT722 driver, LT700, LT723 500Ω output | £2.50 |
| LT30 500mW output | £3.50 |
| 5:1 interstage | £1.50 |
| *Repanco* | |
| T/T3 splitter transformer | £4.00 |
| CH2 5mH RFC | £2.00 |
| *Balanced output transformer* | |
| Vigortronix 600Ω VTX-101-007 | £10 |
| Vigortronix 600Ω VTX-101-3001 | £10 |
| Vigortronix 600Ω VTX-101-3002 | £15 |
| Gardners 150Ω | £10 |
| Reclaimed BBC LL74/MPC nickel core 600Ω | £12 |
| **Low-level 1:1** 600Ω | £4 |
| Reclaimed mic input transformer | £15 |
| *Inductors* | |
| 82µH, 4.7mH, 100µH, 270µH, 10µH, 14µH (low Z) | £0.50 |
| 7-inch ferrite rod with MW and LW windings | £3.00 |

## Special resistors

| | |
|---|---|
| Bourns wire-wound trimmer 5kΩ 3059 JM panel-mount | £2.00 |
| Thermistor RA53, R13 | £4.00 |
| A13 | £2.00 |
| Thermistor CZ1, CZ6 | £1.50 |
| Holco H2 2.2MΩ 1W, 1% | £1.00 |
| Welwyn 1GΩ 2W | £1.00 |
| 5k Bourns 3321H cermet trimmer | £0.50 |

## Potentiometers

| | |
|---|---|
| *Bourns 81* | |
| 25kΩ lin cermet | £2.00 |
| 25kΩ lin, 5kΩ lin conductive plastic | £3.00 |
| *Bourns 91* | |
| 10k dual-gang lin or log | £10.00 |
| *Plessey moulded-track* | |
| 5kΩ log with switch | £3.00 |
| 50kΩ A/log | £3.00 |
| Mil 250kΩ lin dual | £5.00 |
| *Alpha* | |
| 16mm 4.7kΩ A/log | £0.80 |
| 220kΩ A/log | £0.80 |
| 10kΩ lin centre-detent dual-gang | £1.50 |
| *Allen Bradley J series/Honeywell* | |
| 10kΩ lin 1 million cycles | £5.00 |
| Blore Edwards AB 45 dual 5kΩ A/log with switch | £3.50 |
| Alps RK9 dual-gang 5kΩ RD law | £4.00 |
| Alps 50k log tapped motorised stereo | £5.00 |
| BI P260 500kΩ log conductive plastic 1 million cycles | £2.00 |
| Colvern wire-wound 100kΩ or 50kΩ dual-gang 3W | £5.00 |
| Mil Spec hermetic 10Ω | £8.00 |

## Miscellaneous

| | |
|---|---|
| *PCBs* | |
| Cascode Amp (*PE*, Sep 2025) | £2.00 |
| Cascode Amp Power Supply (*PE*, Sep 2025) | £2.00 |
| Pocket Theremin (*EPE*, 1996) | £2.00 |
| Elysian Theremin (*EPE*, 1996) | £6.00 |
| Synth VCF, VCO (*EPE*, 2017) | £3.00 |
| 48V PSU (*EPE*, 2019) | £3.00 |
| SMT dual transistor adapter | £0.40 |
| *Relays* | |
| STC hermetic 4190 24V DPDT | £4.00 |
| *Switches Blore Edwards/Plessey* | |
| MH1 4-pole 3-way | £3.00 |
| Elma 01 2-pole 6-way | £6.00 |
| 3PDT guitar pedal stomp switch | £2.50 |
| *Valves* | |
| EC88, PCL84 | £3.00 |

**Contact**

Jake Rothman
Manchester House
Market Street, Craven Arms
Shropshire SY7 9NN
(visit by appointment)
+44 (0)1597 829102
jrothman1962@gmail.com

Minimum order £5.00 inc post
Quantity discounts negotiable

**Payment**

PayPal, bank transfer, cheques (payable to "J Rothman", UK pounds only)
No VAT payable

**Postage**

Small Jiffy bag £2.99
Small package £4.99
Big boxes and overseas at cost – ask for a quote

# Altium Designer 25

### Review by Tim Blythman

**Altium Designer 25 is the latest version of the EDA (electronics design automation) software that we use for all of our PCB designs. This new version was released late in 2024, and we have spent some time putting it through its paces. Here is what we have found.**

**Altium Designer may not be the cheapest PCB design package,** but it certainly is very capable and becoming quite mature. It has been around for over 30 years now; it can trace its history back to a software package called Protel Autotrax.

Like many software packages, Altium Designer gets frequent updates, including annual major revisions that often introduce new features. I have been using Altium Designer for around ten years now, so I was interested to see what was new compared to the previous major version from 2024, Altium Designer 24. As this is a complex piece of software, I will be concentrating mainly on the change and additions, rather than its core functions like schematic capture, PCB design and simulation.

Renesas Electronics acquired Altium in August 2024. Renesas is a Japanese semiconductor manufacturer that includes the semiconductor operations of Hitachi, Mitsubishi and NEC.

That marks an interesting geographical history for Altium, with Protel Systems Pty Ltd originally founded as an Australian company. For many years, it has been headquartered in San Diego, California.

We're most familiar with Renesas Electronics as the manufacturer of the RA4M1 microprocessor used in the new Arduino Uno R4 microcontroller board. The Raspberry Pi 5 also uses a Renesas power management IC.

## AD25 overview

Altium Designer 25 looks and feels much the same as previous versions, but like many modern applications, it is constantly evolving. This time we are looking at version 25.0.2.

The webpage at www.altium.com/documentation/altium-designer/new lists details of the various version updates and the versions (and subversions) to which they apply. So you can easily see which version has a particular feature.

Some features that are present are disabled by default but can be turned on via the Advanced Settings window of Preferences. Others are available as Extensions, which can be installed from the Extensions and Updates window. Such features may still be at the beta testing stage.

Beta testing means that the feature is essentially complete but not fully tested. It is a good way to get early access to novel features, and there is usually the ability to toggle these features on and off.

Just as with previous versions, it is possible to install multiple versions of Altium Designer alongside each other. You might like to do this to try out the features in a new version without committing to it until you are happy with the changes.

Amongst other videos, the new features are also presented at the Altium Academy YouTube channel (www.youtube.com/@AltiumAcademy). These and other training materials are also available on the Home Page of the Altium Designer application.

## Performance

Altium Designer 25 claims to have much improved performance, especially with large designs. Our designs are typically on the smaller and less complex side, so we're not really able to put this aspect to the test. But we were certainly happy with its responsiveness in the time we've been using it.

This applies to many aspects of Altium Designer, including the schematic and PCB editors, and Draftsman, as well as in collaborative tools like Altium 365 and PCB CoDesign.

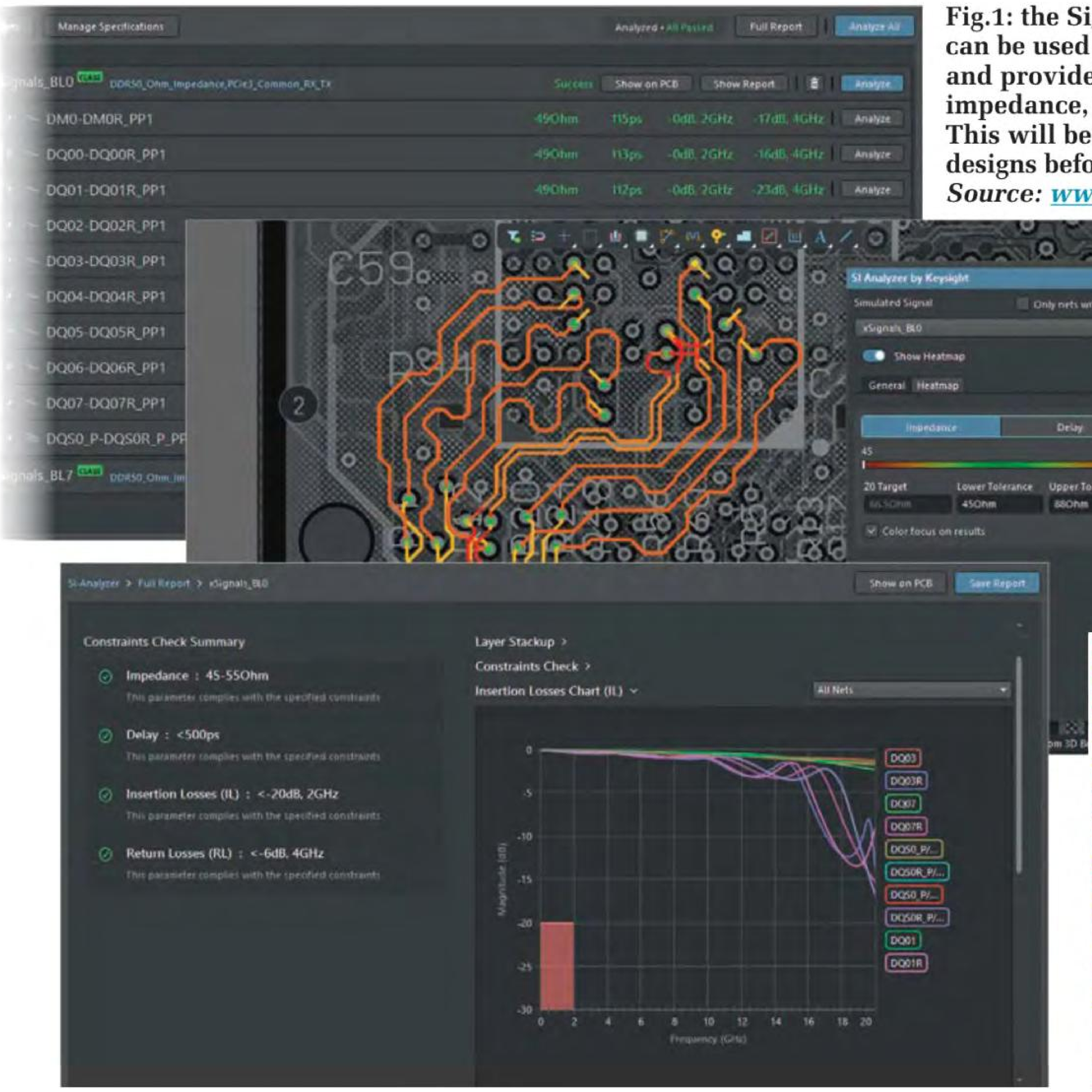
Operations like opening documents, repouring polygons, placing stitching vias and bulk copying and pasting have all been sped up. These improvements have been brought about through better memory management and caching of data where possible.

On a related note, recent versions of Altium Designer (starting with 24.8) use the newer .NET 6 software framework. Previously, Altium Designer used Microsoft's proprietary and now obsolete .NET Framework 4.8. Unsurprisingly, .NET 6 is also faster.

Interestingly, .NET 6 is fully open-source. It is intended to be modular and works across multiple platforms (including Windows, Linux and macOS). Dare we wonder if this is the first step of being able to run Altium Designer on a Linux PC or Mac?

## The SI Analyzer by Keysight

Another extension that sounds quite handy is the Signal Integrity Analyzer by Keysight, although this is another one that we probably won't ever need to use due to our modest designs. As the name suggests, it is intended to


Manage Specifications
Analyzed • All Passed
Full Report
Analyze All
xSignals_BL0 CLASS DDR50_Ohm_Impedance,PCIe3_Common_RX_TX
Success
Show on PCB
Show Report
Analyze
DM0-DM0R_PP1 49Ohm 115ps -0dB, 2GHz -17dB, 4GHz
DQ00-DQ00R_PP1 49Ohm 113ps -0dB, 2GHz -16dB, 4GHz
DQ01-DQ01R_PP1 49Ohm 112ps -0dB, 2GHz -23dB, 4GHz
DQ02-DQ02R_PP1
DQ03-DQ03R_PP1
DQ04-DQ04R_PP1
DQ05-DQ05R_PP1
DQ06-DQ06R_PP1
DQ07-DQ07R_PP1
DQS0_P-DQS0R_P_PP
xSignals_BL7 CLASS DDR50_Ohm_Im
C59
2
SI Analyzer by Keysight
Simulated Signal
Only nets with violations
xSignals_BL0
Show Heatmap
General
Heatmap
Impedance
Delay
45
20 Target
Lower Tolerance
Upper Tolerance
45Ohm
88Ohm
Color focus on results
SI-Analyzer > Full Report > xSignals_BL0
Show on PCB
Save Report
Constraints Check Summary
Layer Stackup >
Constraints Check >
Insertion Losses Chart (IL)
All Nets
Magnitude (dB)
0
-5
-10
-15
-20
-25
-30
0
2
4
6
8
10
12
14
16
18
20
Frequency (GHz)
DQ03
DQ03R
DQ07
DQ07R
DQS0_P/...
DQS0R_P/...
DQS0_P/...
DQS0R_P/...
DQ01
DQ01R

Impedance : 45-55Ohm
This parameter complies with the specified constraints

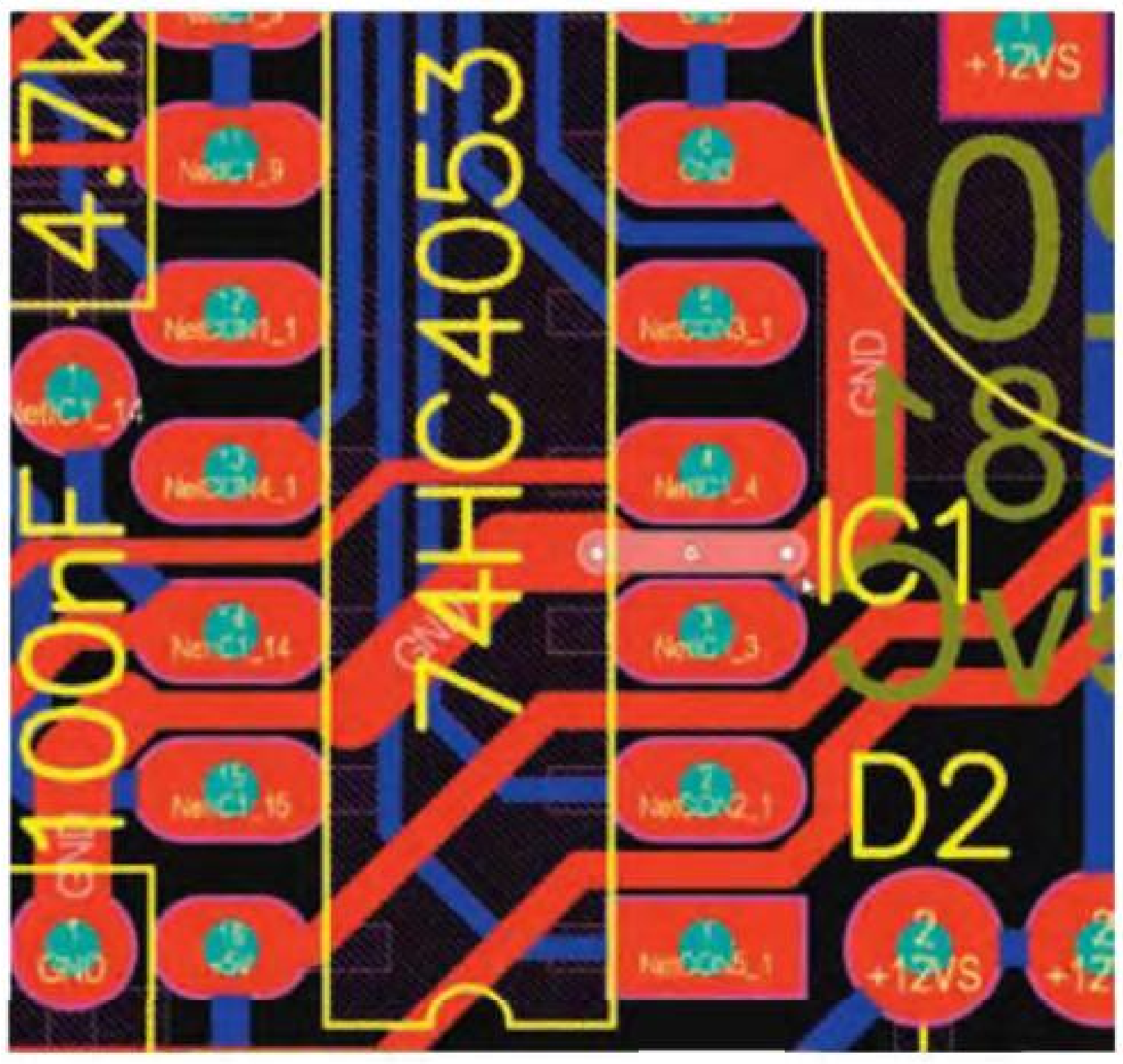
Delay : <500ps
This parameter complies with the specified constraints

Insertion Losses (IL) : <-20dB, 2GHz
This parameter complies with the specified constraints

Return Losses (RL) : <-6dB, 4GHz
This parameter complies with the specified constraints

**Fig.1: the Signal Integrity Analyzer by Keysight can be used post-layout to validate signal integrity and provide checks on parameters such as impedance, delay, insertion loss and return loss. This will be very handy to validate high-speed designs before committing to PCB manufacture.** ***Source:*** [***www.altium.com/documentation/altium-designer/new?version=24#si-analyzer-by-keysight-open-beta-24-10***](www.altium.com/documentation/altium-designer/new?version=24#si-analyzer-by-keysight-open-beta-24-10)

**Fig.2 (below): this trace is 'necked down' to fit through a narrow gap between other pads; we did this manually, using an older version of Altium Designer. The new auto-shrink feature allows this to be done automatically during interactive routing.**


4.7k
100nF
74HC4053
IC1
D2
+12VS
GND
NetIC1_9
NetCON1_1
NetIC1_14
NetIC1_15
NetCON2_1
NetCON3_1
NetIC1_4
NetIC1_3
NetCON5_1

perform signal integrity analysis on high-speed designs. Fig.1 shows the phases of such an analysis.

It can calculate things such as impedance, delay, insertion loss and return loss based on the PCB layout. This can help to validate the PCB before it is committed to manufacture. Signal Integrity Analyzer is currently a beta feature and requires the SI Analyzer by Keysight extension to be installed; there is a free 14-day trial available for this feature.

## Routing

When routing the traces on a board, it is sometimes necessary to use a track narrower than the preferred width to fit through a congested or tight area. This is often described as 'necking down', where the trace is narrowed down to a thin neck just long enough and narrow enough to fit.

Previously, you would have to do this manually, but there is now an option to auto-shrink the width to the minimum you've set.

This will be handy, since manually creating a neck can be a fiddly process, especially if you want the result to be neat. In addition, a new design rule allows the neck to have a maximum specified length, to avoid having too much resistance or increased fragility.

Currently, both these features are in beta and need to be activated in Advanced Settings. The auto-shrink feature is enabled with the "PCB.Routing.EnableAutoShrinking" option, while the neck-down rule follows the "PCB.Rules.RoutingNeckdown" setting.

There is also an option to centre traces when routing. Typically, the auto-router will place traces at the minimum allowed clearance from the nearest track, but spreading the traces out may be preferred. It can also make the routing neater, since the traces will be spread out more evenly.

This is also a beta feature and is set with the "PCB.EnableTraceCentering" advanced setting. Fig.2 shows an example of a trace necking down through a narrow gap on our Thermal Controller PCB, used in a project published in the March & April 2021 issues.

## Single-layer PCBs

For simple designs, a single-layer PCB (with copper on just one side) can be an economical choice, especially for designs on flexible substrates. Large production runs can warrant the savings in eliminating a copper layer where that is feasible.

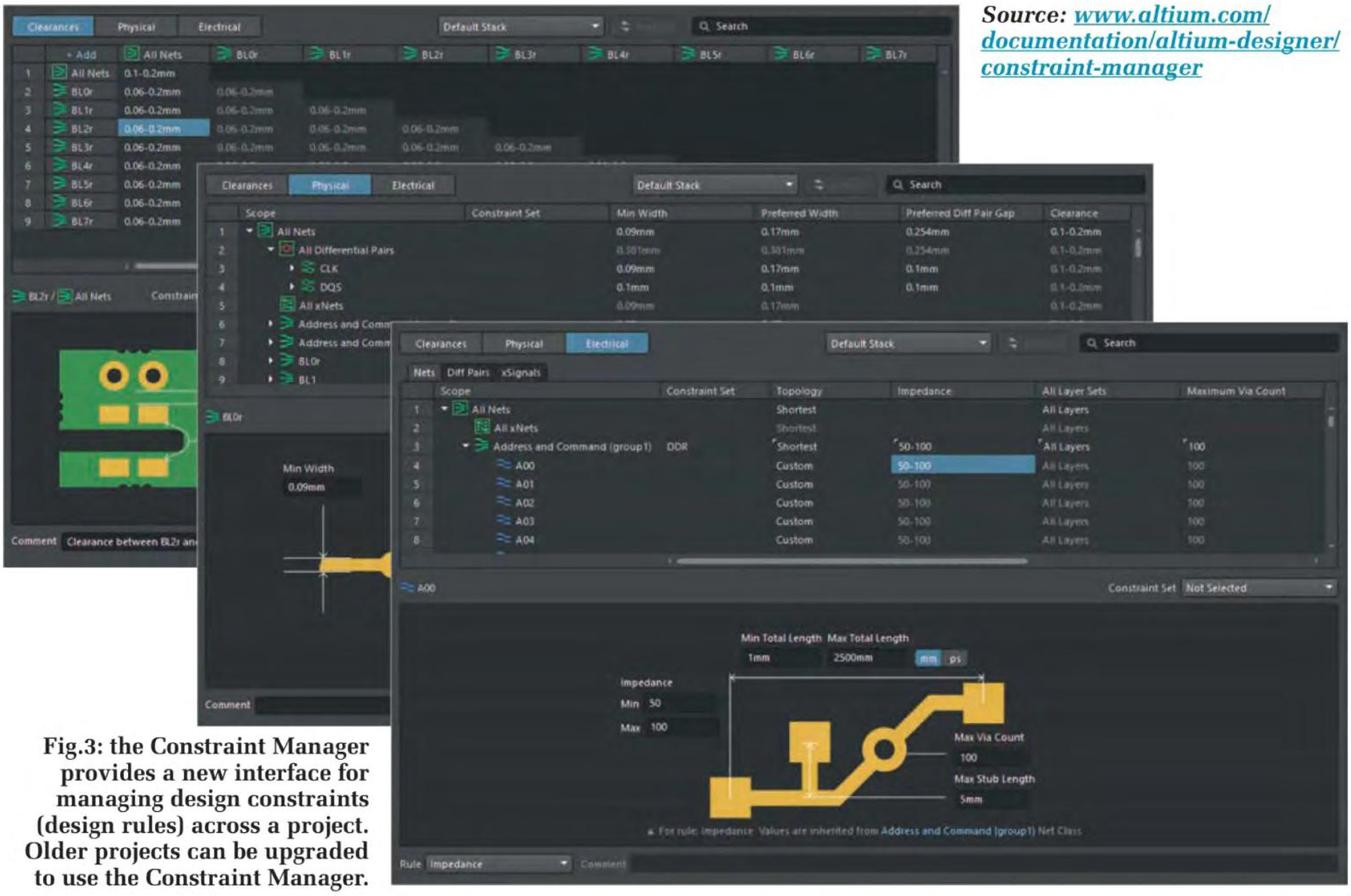
It's now possible to lay out single-layer PCBs by enabling the "PCB.SingleLayerStack.Support" option in Advanced Settings, then removing a copper layer from a two-layer PCB stack.

## Constraint Manager

The Constraint Manager unifies design constraints from both the schematic and PCB layout. It works in place of the PCB Rule and Constraints Editor dialog (Design Rules).

A project can be set up (at creation) to use the Constraint Manager or to use the older Design Rules. There is also a tool that can convert a project from using Design Rules into one that is compatible with the Constraint Manager.

It provides a hierarchical system that is automatically translated into the priority in which rules are applied. Constraint Manager can be enabled by



|  |  | All Nets | BL0r | BL1r | BL2r | BL3r | BL4r | BL5r | BL6r | BL7r |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | All Nets | 0.1-0.2mm |  |  |  |  |  |  |  |  |
| 2 | BL0r | 0.06-0.2mm | 0.06-0.2mm |  |  |  |  |  |  |  |
| 3 | BL1r | 0.06-0.2mm | 0.06-0.2mm | 0.06-0.2mm |  |  |  |  |  |  |
| 4 | BL2r | 0.06-0.2mm | 0.06-0.2mm | 0.06-0.2mm | 0.06-0.2mm |  |  |  |  |  |
| 5 | BL3r | 0.06-0.2mm | 0.06-0.2mm | 0.06-0.2mm | 0.06-0.2mm | 0.06-0.2mm |  |  |  |  |
| 6 | BL4r | 0.06-0.2mm |  |  |  |  |  |  |  |  |
| 7 | BL5r | 0.06-0.2mm |  |  |  |  |  |  |  |  |
| 8 | BL6r | 0.06-0.2mm |  |  |  |  |  |  |  |  |
| 9 | BL7r | 0.06-0.2mm |  |  |  |  |  |  |  |  |

Comment: Clearance between BL2r an

|  | Scope | Constraint Set | Min Width | Preferred Width | Preferred Diff Pair Gap | Clearance |
|---|---|---|---|---|---|---|
| 1 | All Nets |  | 0.09mm | 0.17mm | 0.254mm | 0.1-0.2mm |
| 2 | All Differential Pairs |  | 0.301mm | 0.301mm | 0.254mm | 0.1-0.2mm |
| 3 | CLK |  | 0.09mm | 0.17mm | 0.1mm | 0.1-0.2mm |
| 4 | DQS |  | 0.1mm | 0.1mm | 0.1mm | 0.1-0.2mm |
| 5 | All xNets |  | 0.09mm | 0.17mm |  | 0.1-0.2mm |
| 6 | Address and Comm |  |  |  |  |  |
| 7 | Address and Comm |  |  |  |  |  |
| 8 | BL0r |  |  |  |  |  |
| 9 | BL1 |  |  |  |  |  |

Min Width: 0.09mm

|  | Scope | Constraint Set | Topology | Impedance | All Layer Sets | Maximum Via Count |
|---|---|---|---|---|---|---|
| 1 | All Nets |  | Shortest |  | All Layers |  |
| 2 | All xNets |  | Shortest |  | All Layers |  |
| 3 | Address and Command (group1) | DDR | Shortest | 50-100 | All Layers | 100 |
| 4 | A00 |  | Custom | 50-100 | All Layers | 100 |
| 5 | A01 |  | Custom | 50-100 | All Layers | 100 |
| 6 | A02 |  | Custom | 50-100 | All Layers | 100 |
| 7 | A03 |  | Custom | 50-100 | All Layers | 100 |
| 8 | A04 |  | Custom | 50-100 | All Layers | 100 |

A00

Constraint Set: Not Selected

Min Total Length: 1mm
Max Total Length: 2500mm

Impedance
Min: 50
Max: 100
Max Via Count: 100
Max Stub Length: 5mm

For rule Impedance. Values are inherited from Address and Command (group1) Net Class

Rule: Impedance

**Source: *www.altium.com/documentation/altium-designer/constraint-manager***

**Fig.3: the Constraint Manager provides a new interface for managing design constraints (design rules) across a project. Older projects can be upgraded to use the Constraint Manager.**

setting the "System.ConstraintManager" option in Advanced Settings. Fig.3 shows some views of the Constraint Manager.

## Importing

Occasionally, we have to deal with contributed PCBs that have been designed using a different EDA tool, and sometimes we need to change them. This might be as simple as making a small change to the silkscreen markings or could involve a major revision of the copper routing layers.

Some changes can be made by directly editing the Gerber files, but having access to fully editable PCB design files is better for many reasons. Firstly, that makes it possible to run design rule checks to validate that any changes do not cause a manufacturing issue like shorted traces. It then becomes possible to make further revisions if needed in the future.

Whatever the reason, this means that we need the ability to import designs from other EDA tools into Altium Designer so that they can be turned into native Altium Designer files, such as SchDoc schematic files or PcbDoc PCB files.

The latest version of the importer now works with KiCad designs from KiCad version 7 or 8. This is available as an extension known as the KiCad Importer Extension and can be found in Extensions and Updates.

Their website at **https://pemag.au/link/ac41** has more information on importing designs from other tools. Fig.4 shows a screenshot from our installation of the Software Extensions that are on offer.

We have also previously used the free online Altium 365 Viewer, which is at **www.altium.com/viewer**

This now supports KiCad files, and we had no trouble using it to view some KiCad PCB files that we found online.

## Wire bonding

Altium Designer can be used for designing much more than just traditional planar substrate PCBs. We were already aware of Altium Designer's ability to work with flexible and hybrid (rigid-flex) PCBs, printed electronics and 3D-printed substrates using the 3D-MID (three-dimensional mechatronic integrated device) process. Another example is the recent Harness Designer.

One of the technologies supported by Altium Designer 25 is COB (chip-on-board) using wire bonding. COB involves bare silicon chip dies being bonded directly to a PCB. Connections are made from pads on the die (die pads) to the bond finger pads on the PCB by means of bond wires.

The bond wires used in COB applications are much the same as the bond wires used to connect a silicon chip to its leadframe in a traditionally packaged integrated circuit. These are very fine wires of a metal such as gold, copper or aluminium that are welded to their pads using heat, pressure or ultrasonic energy.

The process is typically performed by an automated robotic system. You can see some images of a COB design at **www.altium.com/documentation/altium-designer/wire-bonding** while the photos at upper right shows two examples of silicon dice bonded to a PCB.

Creating a COB design involves adding a Die layer and a Bond wire layer to a PCB document or library. A complete footprint 'package' including die pads, bond finger pads, and bond wires can be created, in which case the

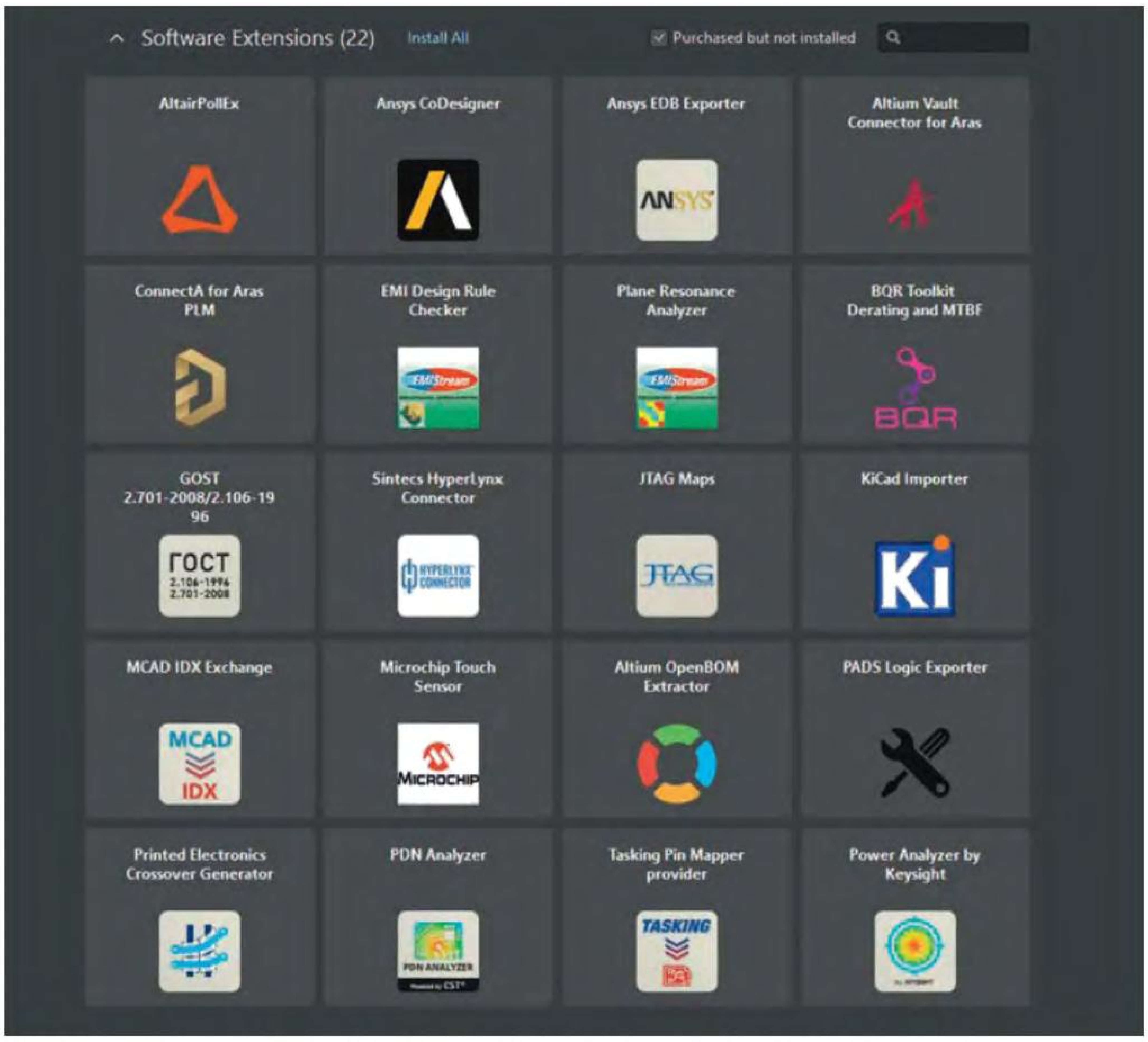
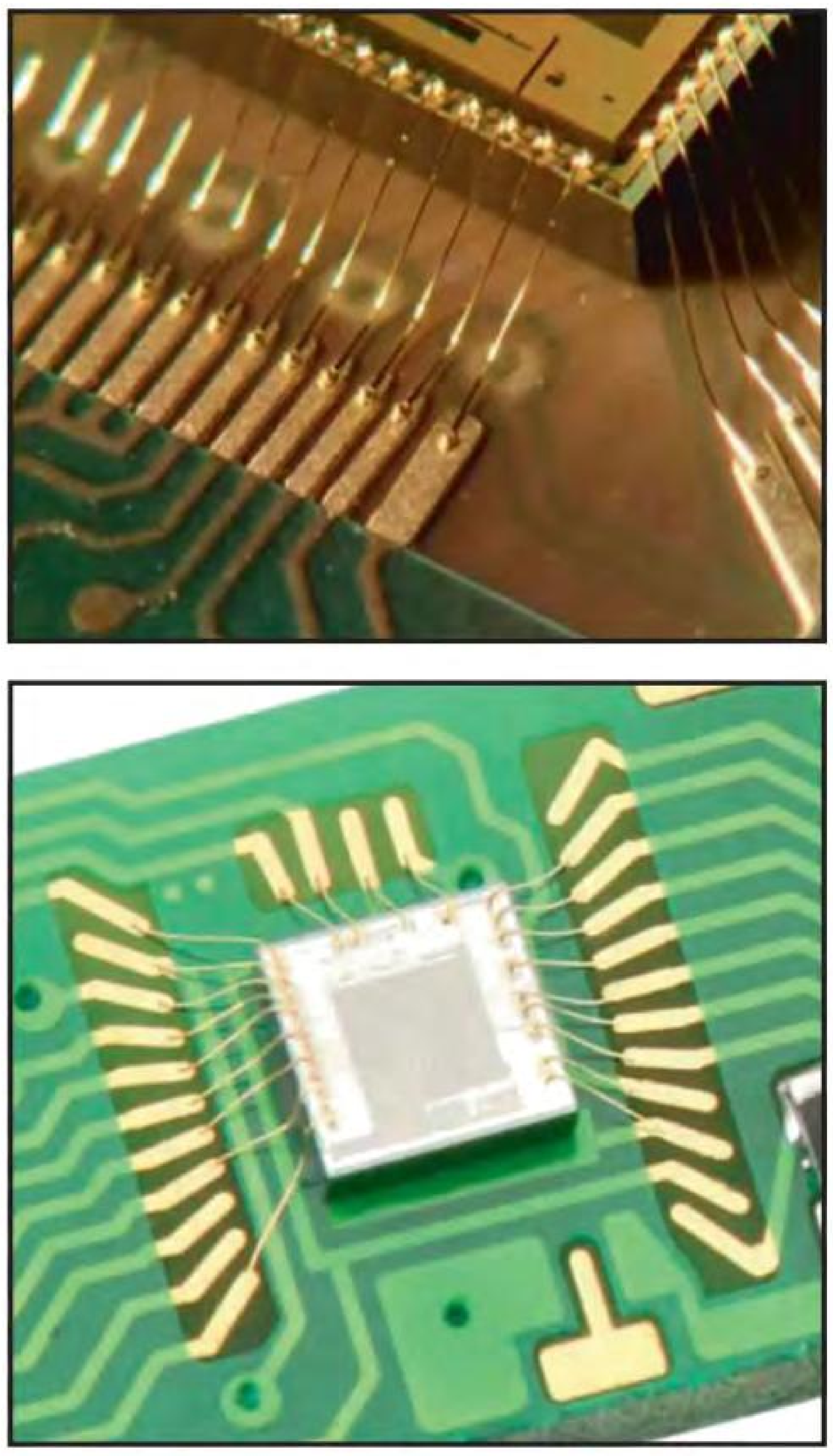

Software Extensions (22)
Install All
Purchased but not installed
AltairPollEx
Ansys CoDesigner
Ansys EDB Exporter
Altium Vault Connector for Aras
ConnectA for Aras PLM
EMI Design Rule Checker
Plane Resonance Analyzer
BQR Toolkit Derating and MTBF
GOST 2.701-2008/2.106-19 96
Sintecs HyperLynx Connector
JTAG Maps
KiCad Importer
MCAD IDX Exchange
Microchip Touch Sensor
Altium OpenBOM Extractor
PADS Logic Exporter
Printed Electronics Crossover Generator
PDN Analyzer
Tasking Pin Mapper provider
Power Analyzer by Keysight

**Fig.4: numerous Altium Designer extensions can be installed from the Extensions and Updates tab. This shows just some of the extensions available on our system.**



**Two examples of chips bonded directly to PCB pads. *Source:* *www.rocket-pcb.com/rocket-pcb-wholesale-wire-bonding-technology-bulk-fabrication-for-electronics***

bond finger pads can be simply routed to on the copper layers.

Alternatively, a die with die pads can be placed alongside separate bond finger pads. The bond wires are then placed manually. With the current version, wire bonding can be enabled with the "PCB.Wirebonding" option in the Advanced Settings dialog. For manufacture, a wire bonding table report can be generated.

As you can see from Fig.5, Altium Designer's 3D rendering allows you to see all aspects of a wire-bonded design. For example, you can visually check that crossed bond wires pass at different heights to avoid collisions. We find the 3D views invaluable for making sanity checks on our PCB layouts.

## Licensing

There are three different subscription levels available for Altium Designer: Standard, Pro and Enterprise. They differ in the number of features that are included and more information is available at **www.altium.com/altium-designer/licensing**

## Resources

Altium offers numerous resources to allow users to make the best use of Altium Designer. You can view a recent webinar (recorded in November) at **https://pemag.au/link/ac42**

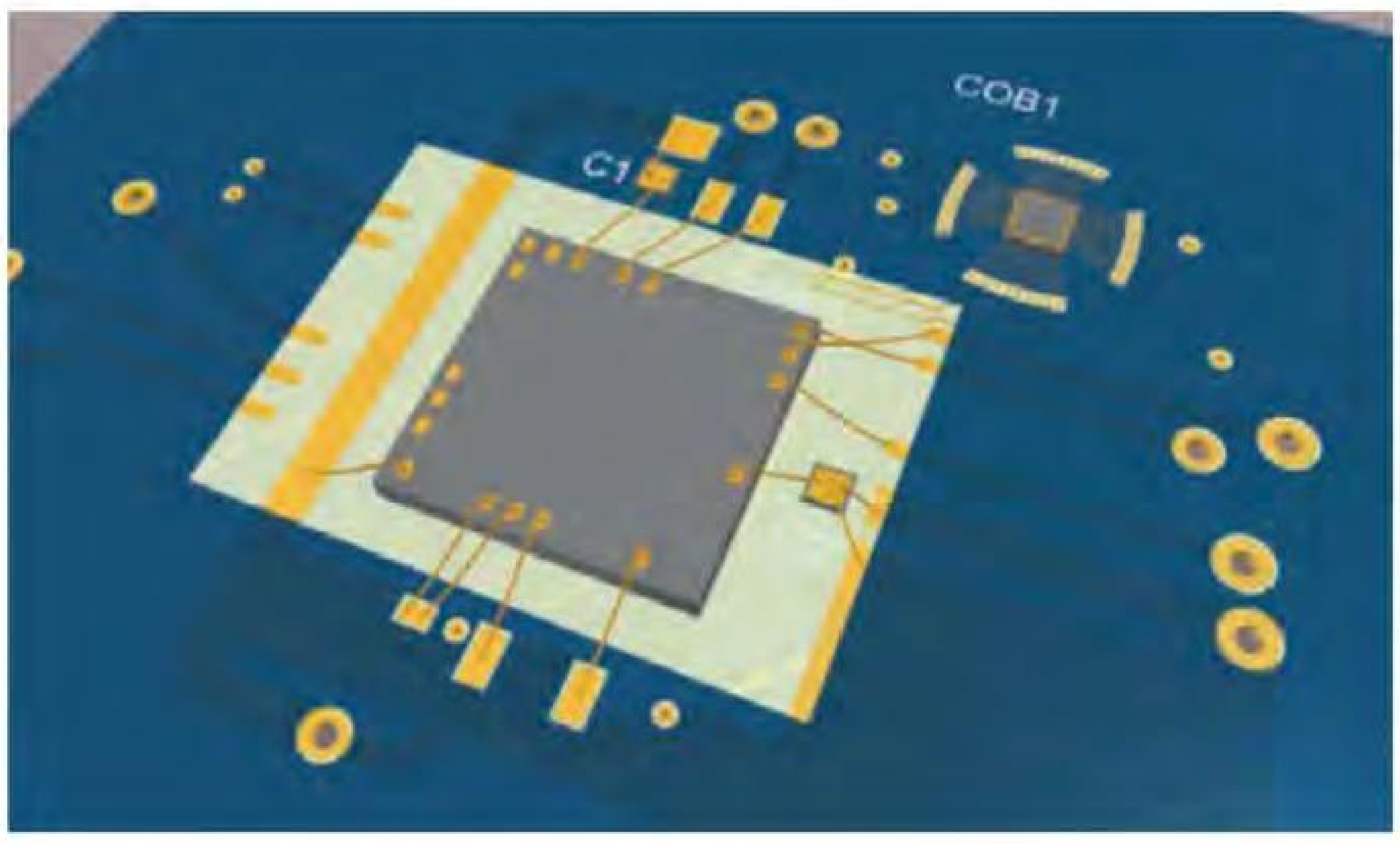


**Fig.5: Altium Designer 25 now allows wire bonding between silicon dies and PCBs. This image shows an example of two COB (chip on board) dies bonded to the PCB below. *Source:* *www.altium.com/documentation/altium-designer/wire-bonding#placing-wire-bonds-in-a-pcb***

In addition to covering the important aspects of a feature (in this case, Single-layer PCBs), it follows with a live demonstration of how to use that feature within Altium Designer.

Altium also has a free trial offer for new users. It allows you to use a fully featured trial version of Altium Designer for 15 days; see **www.altium.com/altium-designer/free-trial**

## Conclusion

Altium continues to make incremental updates to Altium Designer. While there are some bigger features that we would struggle to use fully, such as wire bonding, they are no doubt useful for larger organisations. It is good to see that they continue to improve usability and work on basic features such as routing.

We hope that the switch to the .NET 6 framework is the beginning of cross-platform support. We're especially keen to be able to use Altium Designer on Linux, since so many of the other programs that we use already allow that.

For more information on Altium Designer, visit **www.altium.com/altium-designer** PE

**Project by Stefan Keller-Tuberg**

This device will disconnect a load from its power supply if the voltage is reversed or too high. It also disconnects the load if it draws current above the adjustable trip level. Its dual-rail support means it can work with devices like audio amplifiers with a split (positive and negative) supply.

FlexEFuse
SC

# Dual-Rail Load Protector

**In May 2025, we published three DC Supply Protectors** that guard against reversed or excessive supply voltages, but they could only handle a single-rail supply.

This design provides even more functions, extends the reverse/overvoltage protection to split rails and adds adjustable current limits with automatic or manual resetting.

If you've built something that uses flying power leads, you may already have had a close call mixing up polarities. Or have you ever forgotten to check that you're using the right supply to power a device? If any of these ring true, this design might help avoid a catastrophe by introducing power supply protection. It's so versatile that you'll think of many applications for it.

The overvoltage cut-out levels can be set between ±5V and ±19V (or 5-38V for the single-rail version). If the supply overshoots the protection level, this device will rapidly interrupt it.

The overcurrent thresholds are set by a current sense resistor and trimpot. The sense resistance is chosen so the voltage drop is approximately 50mV at the nominal protection level. The trimpot range permits adjustment from zero up to twice the nominal current level.

When the current limit is reached, it interrupts the offending power rail by turning it off completely, similar to a fuse blowing. This minimises the chance of damage due to a fault compared to simply holding the current at the threshold by reducing the voltage, as a current-limited bench supply would do.

Also, if the device is unattended when it fails, interrupting rather than limiting the power delivered could help avoid an even larger disaster.

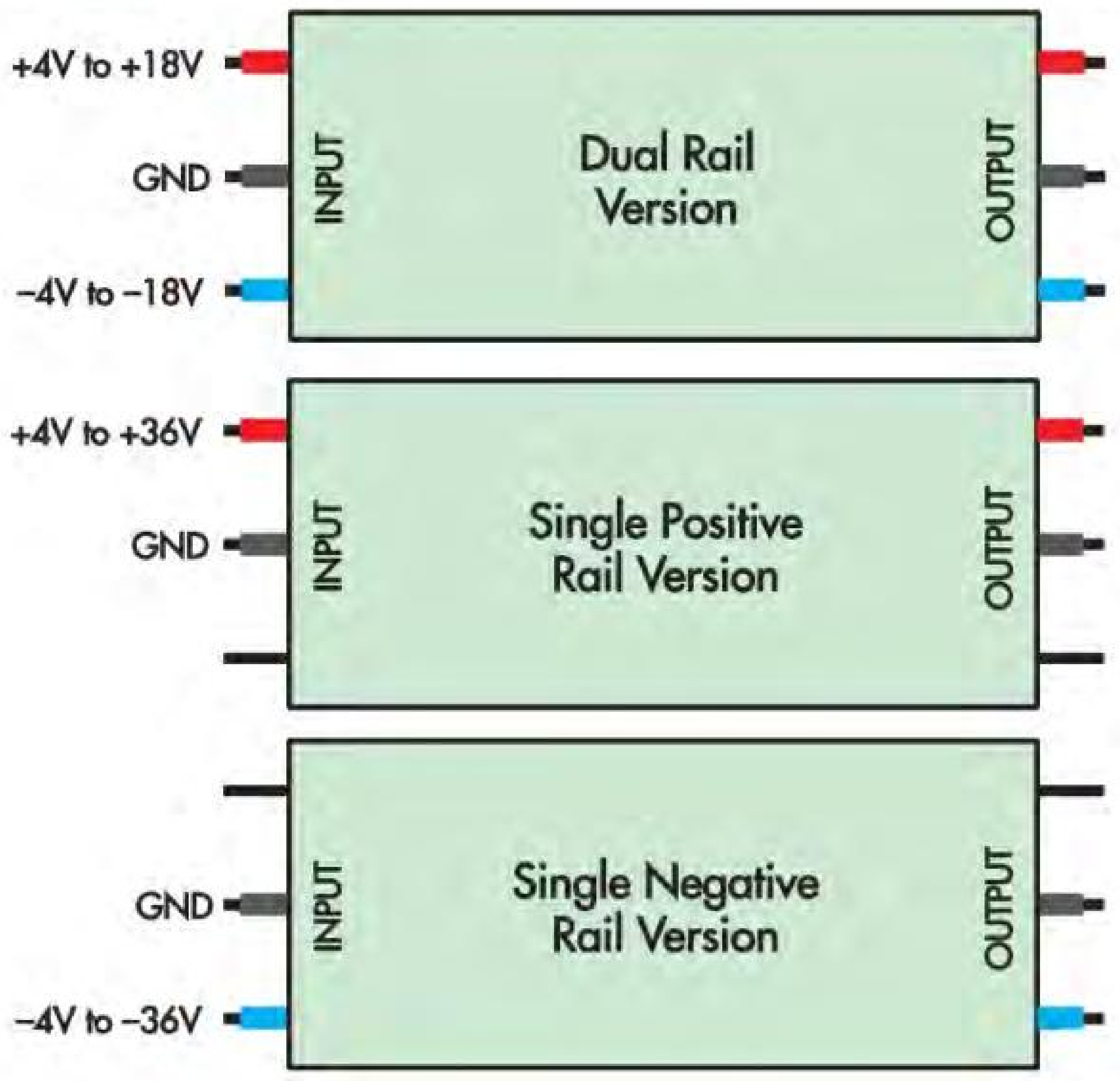
It can be set so that when the overcurrent circuit trips, it will automatically reconnect after a two-second delay or require manual intervention (a button press). If your dual-rail application is asymmetric, you can set different overvoltage and overcurrent thresholds for each rail.

Depending on the Mosfets used, it can handle up to 4-7A per rail without heatsinking. Adding heatsinks to the Mosfets will allow them to handle more, up to 10A for the higher-current Mosfets specified.

Due to the design's modularity, you only need to populate the required features. To start with, you can equip it to suit single or dual supply rails. Configuring it for a single rail saves a few components and doubles the single supply maximum voltage to 36V.

If your device to be protected has more than one positive or more than

**Fig.1: some of the different ways the Supply Protector can be used. If the maximum voltage of ±18V for the dual-rail version is not enough for your application, you can stack two boards to double that, as shown on the far right.**



**Features & Specifications**

- Voltage range: 4-36V DC or ±4-18V DC (±4-36V DC with two boards)
- Over-voltage cut-out: 5-38V DC or ±5-19V DC (±5-38V DC with two boards)
- Voltage withstand: up to ±60V at either input or across both inputs ◆
- Current capability: 7A+ without heatsinking (more with heatsinks) ◆
- Voltage insertion loss: typically <300mV @ 10A ◆
- Over-current protection: disconnects rails independently if current draw exceeds a set threshold
- Over-current reset: automatically after two seconds or manually via pushbutton

◆ *exact values depend on parts used*

one negative rail, you can common the grounds and use two or more of these boards to protect them all, including dual-rail applications operating up to ±36V, as shown in Fig.1.

You can leave some components off if you don't need overcurrent protection. You can also leave off the overvoltage sections if you don't want that feature. The circuit is arranged in three sections, each supporting one or two power rails.

## Reverse polarity protection

The first section of the circuit, shown in Fig.2, uses Mosfets Q1 & Q2 like 'ideal diodes'. They have a very low voltage drop when forward-biased but a high impedance when reverse-biased.

If you accidentally connect the input voltages with the wrong polarity, the internal body diodes of Mosfets Q1 and Q2 will be reverse-biased, and no current can flow. Q1 and Q2 remain off, protecting all the downstream components from the abnormal condition. The specified Mosfets have reverse voltage ratings up to 60V, offering plenty of protection against accidental power supply reversal.

However, without protection, the Mosfets could be damaged by gate-to-source voltages exceeding 20V. Zener diodes ZD1 and ZD2 ensure that the voltage between the gate and source of each Mosfet cannot exceed 15V. Other Mosfets in the design have similar protections.

When the input voltage polarities are correct, the internal diodes of Q1 & Q2 are forward-biased. As current starts to flow, 47kΩ resistors pull the Mosfet gates to ground, so they switch on. As the gate bias exceeds 2-4V and the Mosfet channel resistance drops, the internal protection diodes will be shunted, so very little voltage will be lost across the Mosfets.

The Supply Protector's minimum voltage rating of 4V is because that is the minimum voltage at which the Mosfets used are guaranteed to switch on and conduct sufficient current.

## Over-voltage protection

The following section deals with over-voltages. Zener diodes ZD3 and ZD6 set a fixed value for each rail's protection threshold. The knee voltage for 1W zener diodes rated above 5.6V occurs around 3.5-5% below the nominal zener voltage. As the supply voltage reaches this level, they will start to break down. Lower voltage zener diodes have a more rounded knee, so the difference from nominal can be larger.

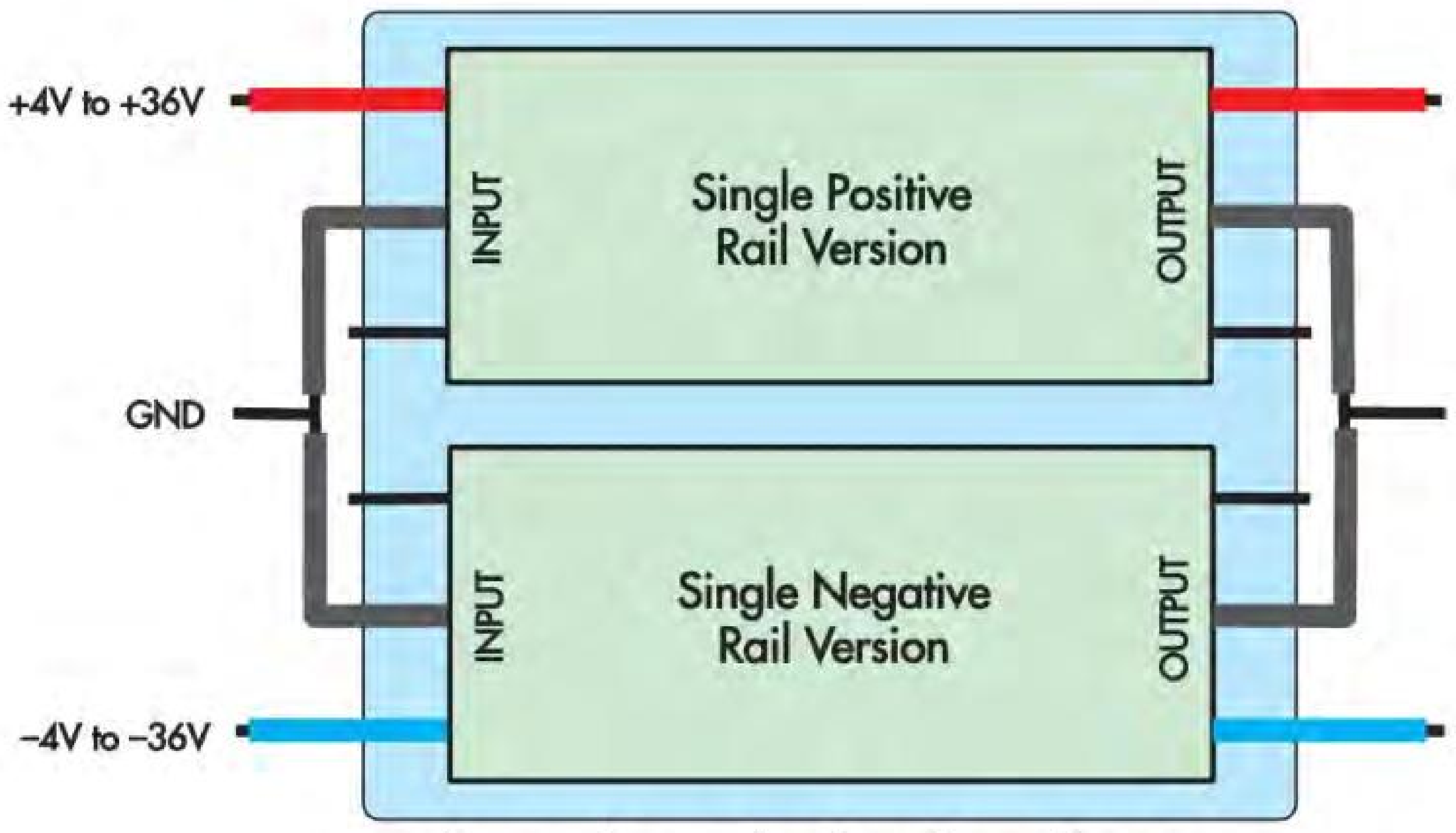
When enough current flows to develop 0.6V at the gate of the associated SCR, it will trigger and switch off either Mosfet Q4, in series with the positive rail, or Mosfet Q5 in the negative rail, disconnecting and protecting the downstream circuitry and the load.

The SCR will remain latched until the supply voltage is removed. Providing the applied voltage remains below the Mosfet specification (55V or 60V), the unit will tolerate the condition indefinitely, and the device you're protecting will stay safe.

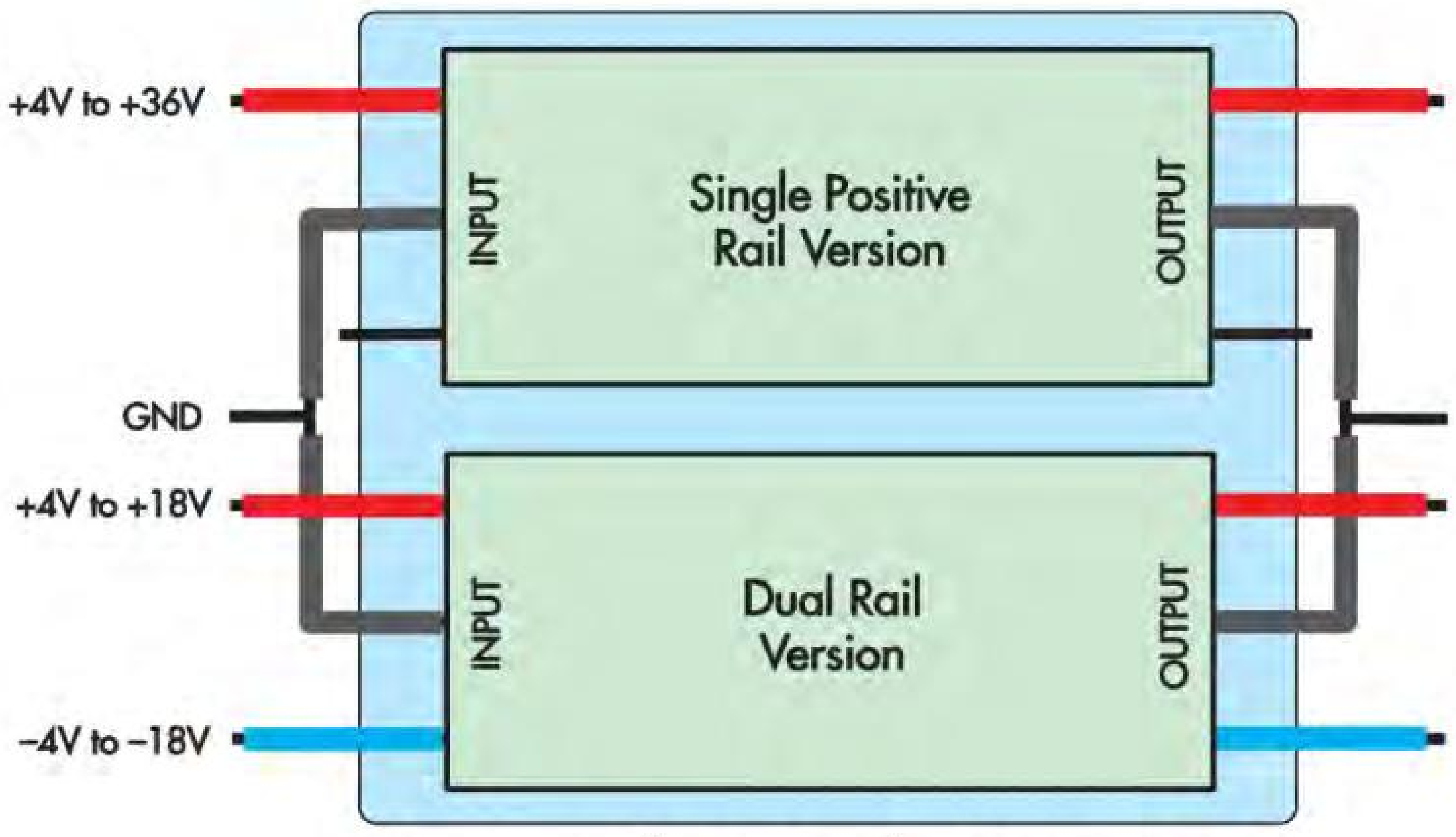
Most applications won't require fine overvoltage threshold adjustment, so you can simply set it by selecting the nearest zener. Two extra diodes labelled D4 and D5, in series with the zeners, allow the threshold to be tweaked. Usually, they are replaced with wire links, but if required, regular or schottky diodes can be fitted to increase the overvoltage trip thresholds by 0.3V (SB140/1N5819) or 0.6V (1N4004).

Op amp IC1 has an absolute maximum limit of 40V, the highest overvoltage threshold supported. In practice, the trip points should be no more than ±19V for one dual-rail device or 38V for a single-rail version, giving a small safety margin.

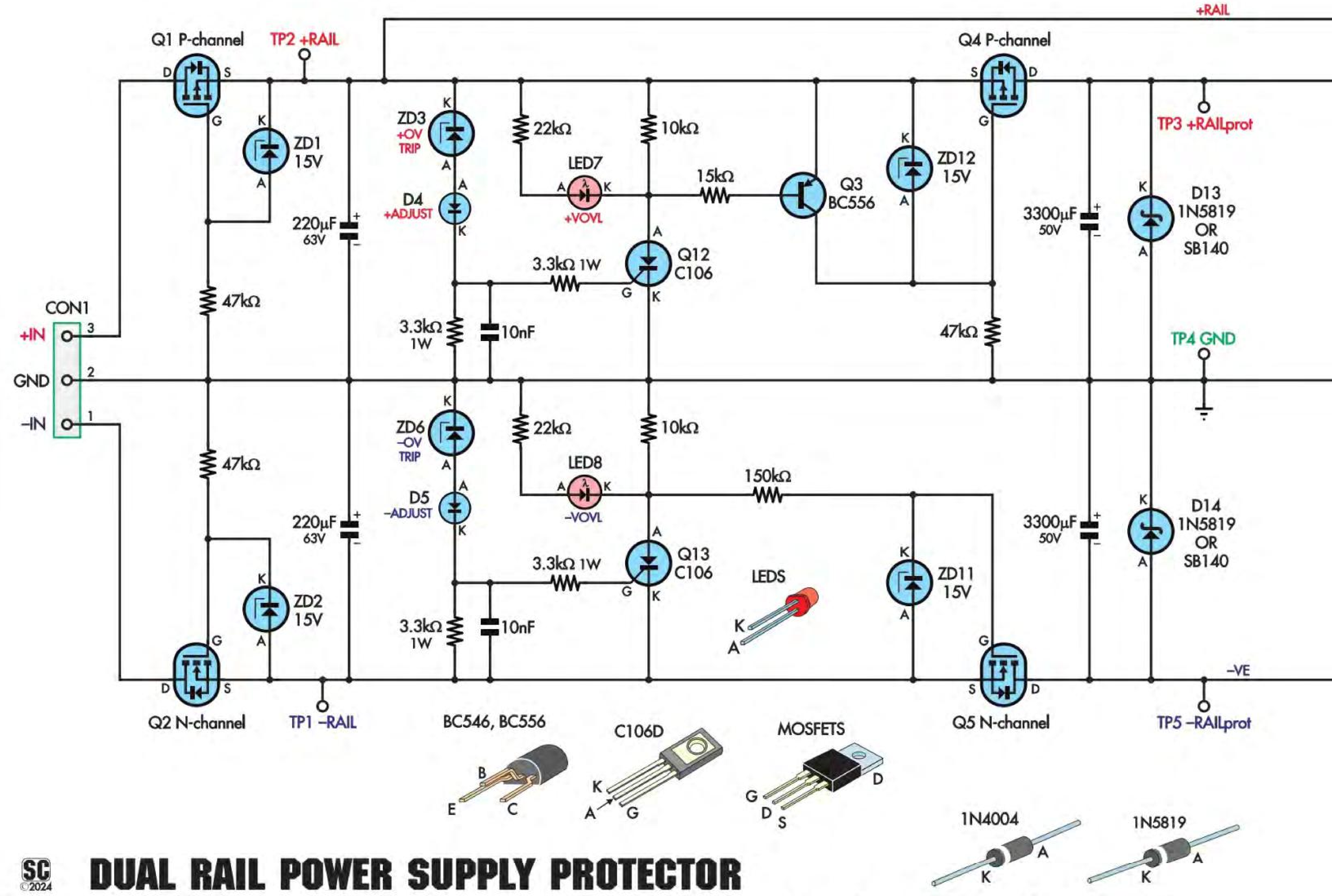
The 220µF and 3300µF electrolytic capacitors are to counteract the effects of power source inductance. At switch-on, many devices cause a momentary current surge as the supply bypass



Higher voltage dual-rail applications



Common the grounds to protect any combination of positive & negative rails


+RAIL
Q1 P-channel
TP2 +RAIL
Q4 P-channel
TP3 +RAILprot
ZD1 15V
ZD3 +OV TRIP
D4 +ADJUST
22kΩ
10kΩ
LED7 +VOVL
15kΩ
Q3 BC556
ZD12 15V
3300µF 50V
D13 1N5819 OR SB140
220µF 63V
3.3kΩ 1W
Q12 C106
47kΩ
CON1
+IN
GND
–IN
3.3kΩ 1W
10nF
TP4 GND
ZD6 –OV TRIP
D5 –ADJUST
LED8 –VOVL
150kΩ
Q13 C106
LEDS
ZD11 15V
D14 1N5819 OR SB140
ZD2 15V
Q2 N-channel
TP1 –RAIL
Q5 N-channel
–VE
TP5 –RAILprot
BC546, BC556
C106D
MOSFETS
1N4004
1N5819
SC ©2024

**DUAL RAIL POWER SUPPLY PROTECTOR**

**Fig.2: the Supply Protector circuit has mostly independent positive and negative sections with three stages each. The first is reverse polarity protection (using Mosfets Q1 & Q2), followed by overvoltage protection (Mosfets Q4 & Q5), then overcurrent protection (Mosfets Q10 & Q11). The only sections shared between the positive and negative rails are the half-supply generator (IC1d), reset oscillator (IC1c) and reset switch.**

capacitors charge. This high current pulse can interact with inductances in the wiring etc, causing 'ringing' (oscillation), which causes an overshoot voltage to appear on the affected power rail, sometimes a significant one.

One of my test supplies caused damped oscillations with a frequency of around 2MHz, resulting in a peak overshoot voltage of around 50% above the nominal supply. This persistently tripped the overvoltage protection at power-on.

The electrolytic capacitors dampen power-on overshoot to avoid false overvoltage trips. In severe cases, you may need to increase the value of the 220µF parts, although they should be sufficient for most cases. It is usually more severe with a longer input power cable.

## Overcurrent protection

The third section of the circuit provides overcurrent protection. The load current is monitored by the voltage drop across the '+sense' and '-sense' resistors. For the positive rail, LED15, VR1 and one '+bias' resistor set an adjustable reference voltage at pin 3 of IC1a that is a couple of volts below the +ve rail voltage.

LED17 and the other '+bias' resistor create another voltage at IC1's pin 2 that varies with the '+sense' voltage drop. IC1a compares these voltages; its output is low when the sensed current is below the setpoint, so Mosfet Q10 is usually on. If the current setpoint is exceeded, IC1a's output goes high, switching off Q10 and disconnecting the load, while also lighting overcurrent indicator LED21. Op amp IC1b, Mosfet Q11 and LED22 function similarly for the negative rail.

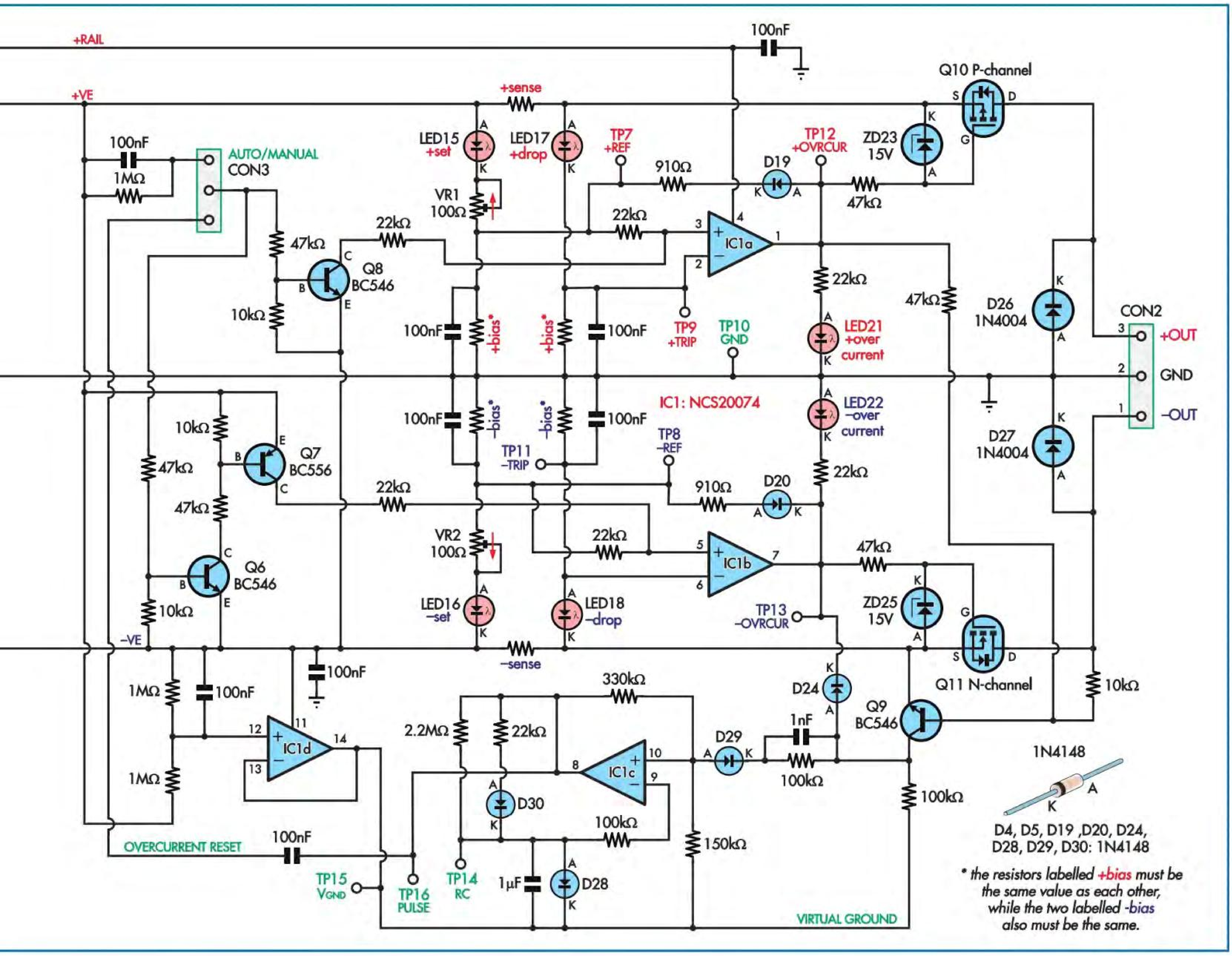
The purposes of LED15 and LED17 aren't to emit light; they provide consistent voltage drops so the op amp inputs remain within the chip's common-mode range, which does not go up to the positive rail. The fact that LEDs have a higher voltage drop than a regular silicon diode (around 1.8V rather than 0.7V) is useful in this application.

The voltage across the '+sense' resistors is approximately 50mV at the nominal overcurrent trip point. VR1 is for fine-tuning; its 100Ω value means that with 1mA flowing through it, a full trimpot rotation will cover twice the nominal voltage range expected across '+sense'.

Note that the '+set' LED (LED15) usually goes out when the overcurrent LED (LED21) lights. However, there are cases where the current is near the overcurrent set point where both could light. So if you notice that, it's normal.

The overcurrent protection only interrupts the rail experiencing the overload. When that happens for the

# Dual-Rail Load Protector


+RAIL
100nF
+VE
+sense
Q10 P-channel
S
D
G
100nF
1MΩ
AUTO/MANUAL
CON3
LED15
+set
LED17
+drop
TP7
+REF
TP12
+OVRCUR
ZD23
15V
K
A
910Ω
D19
47kΩ
VR1
100Ω
22kΩ
22kΩ
3
4
1
2
IC1a
47kΩ
Q8
BC546
C
B
E
10kΩ
22kΩ
47kΩ
D26
1N4004
CON2
3
+OUT
2
GND
1
–OUT
100nF
+bias*
+bias*
100nF
TP9
+TRIP
TP10
GND
LED21
+over
current
100nF
–bias*
–bias*
100nF
IC1: NCS20074
LED22
–over
current
D27
1N4004
10kΩ
Q7
BC556
TP11
–TRIP
TP8
–REF
47kΩ
47kΩ
22kΩ
910Ω
D20
22kΩ
VR2
100Ω
22kΩ
5
7
6
IC1b
47kΩ
Q6
BC546
ZD25
15V
10kΩ
LED16
–set
LED18
–drop
TP13
–OVRCUR
Q11 N-channel
–VE
–sense
D24
10kΩ
100nF
330kΩ
Q9
BC546
1MΩ
100nF
12
11
14
13
IC1d
2.2MΩ
22kΩ
1nF
D29
8
10
9
IC1c
100kΩ
1N4148
1MΩ
D30
100kΩ
100kΩ
D4, D5, D19 ,D20, D24,
D28, D29, D30: 1N4148
OVERCURRENT RESET
100nF
150kΩ
TP15
VGND
TP16
PULSE
TP14
RC
1µF
D28
VIRTUAL GROUND
* the resistors labelled +bias must be the same value as each other, while the two labelled -bias also must be the same.

positive rail, D19 provides a feedback path to latch the state even after Q10 interrupts the current and the '+sense' voltage drop falls back to zero. It will remain off until the condition is reset by NPN transistor Q8 switching on and pulling pin 3 of IC1a below the pin 2 level.

When the output of IC1a goes high, another NPN transistor (Q9) inverts the transition to create a falling edge. This is combined with any falling edge from IC1b by diode D24. These are AC-coupled to IC1c by a 1nF capacitor and diode D29, which works as an overcurrent reset monostable. A two-second delay is provided by the 1μF capacitor and 2.2MΩ resistor. The 100kΩ resistor at pin 9 of IC1c prevents damaging input currents when pins 9 and 10 differ by more than 5.5V.

When the monostable times out (if enabled) or the reset pushbutton is pressed, Q6, Q7 and Q8 temporarily shift the voltage levels at the inputs of IC1a and IC1b. This forces them out of their latched states, re-enabling Mosfets Q10 and Q11.

If the auto-reset feature is enabled, voltage to the load will be restored two seconds after it trips. If the overcurrent condition persists, the trigger-delay-restore process will repeat indefinitely every two seconds (or until the fault clears).

This monostable arrangement requires a reference voltage at the midpoint of the IC's power supply. We could have used the GND line for this reference, but that would mean the circuit would only work with symmetrical dual rails, reducing its flexibility. So IC1d synthesises a mid-rail voltage (halfway between +ve and -ve) that self-adjusts without needing different configurations.

The 100nF capacitor and 1MΩ resistor connected to header CON3 ensure that resets are only momentary, even if the pushbutton is held down. This quickly rearms the overcurrent protection while preventing the output from being held on continuously if excessive current continues to flow. The 100nF capacitor at IC1c's output works similarly for monostable-initiated resets.

We don't care about the brightness of LED15-LED18 since, as mentioned, they are not indicators. However, it does matter for LED7, LED8, LED21 and LED22. The circuit shows 22kΩ current limiting resistors for them, suiting high-brightness LEDs. If using regular LEDs, reduce the values to around 5.6kΩ for more current.

Alternatively, if they're too bright (which may happen with higher-voltage single-rail applications), increase the series resistances.

Diodes D26 and D27 across the output terminals are normally reverse-biased. These protect the circuit from inductive loads or long output power leads. Any inductance in these can cause a reverse voltage spike when the load current is interrupted (as can certain capacitor configurations in the load). These diodes will safely dissipate that energy.

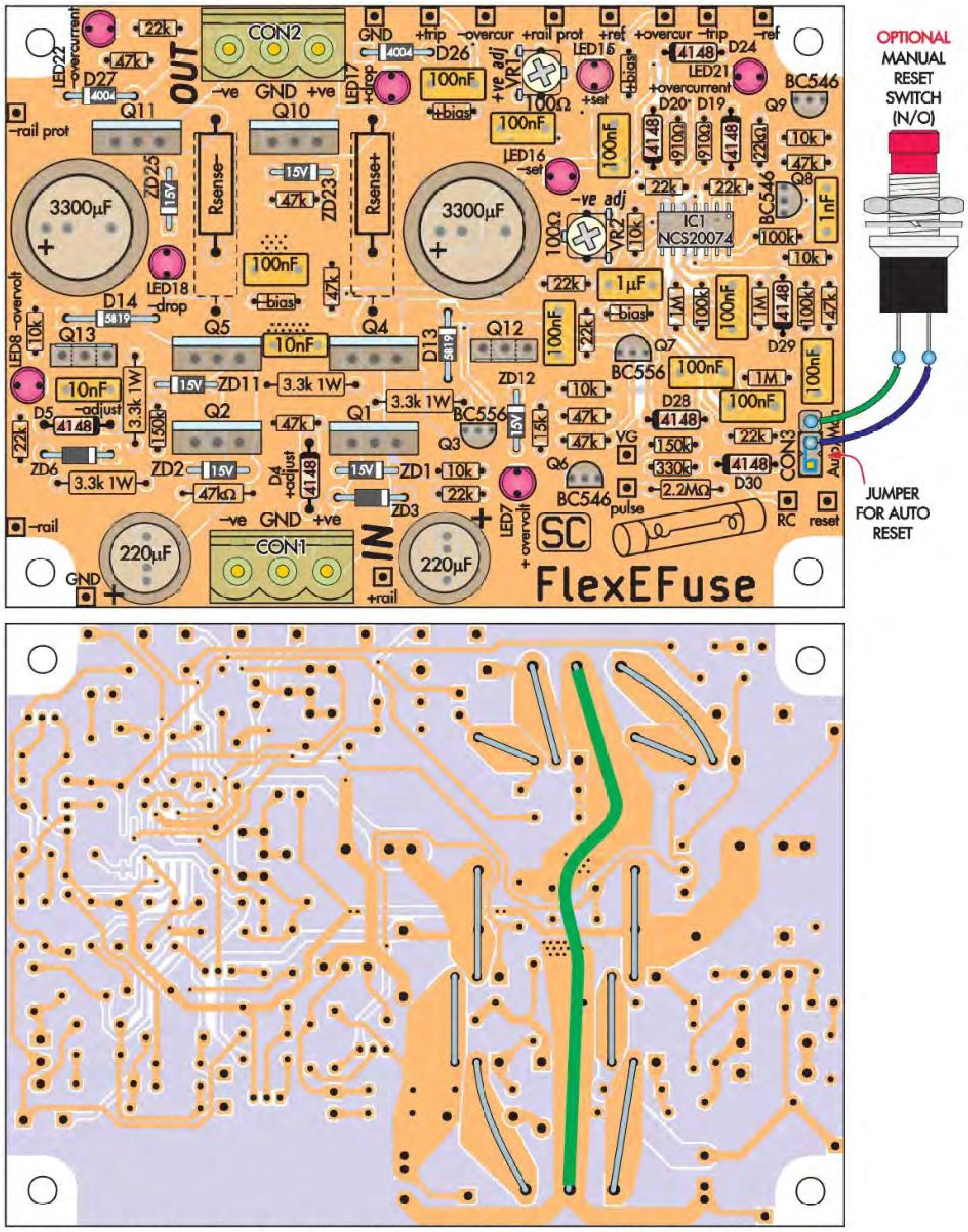
## Component selection

The PCB is designed for miniature ⅛W resistors, 3.5mm long. You can get them from element14, DigiKey or Mouser. You can use more common ¼W resistors, but you will need to stand them up (at least partially).

The current sense resistors will typically be below 0.1Ω. element14, DigiKey and Mouser have wide ranges of low-value 'current sense' resistors, many of which will be suitable, even if their power ratings are higher than necessary. Alternatively, parallel two or more resistors to create a lower resistance. The PCB holes are large enough for the leads of current-sense resistors or multiple regular resistors.

The parts list gives information on supported capacitor lead pitches (although you can bend them if you have to) and suggested Mosfet types. However, many more suitable ones will be available.

If substituting other Mosfets, pay particular attention to their maximum on-resistance, $V_{GS}$ threshold voltage and reverse breakdown voltage. Particularly for P-channel Mosfets, cost-effective options with a low on-resistance aren't common. The higher the maximum on-resistance, the hotter the Mosfets will run.



**Figs.3 & 4: the dual-rail version of the Supply Protector uses all the parts on the PCB, although some sections can be omitted. Parts that can be left off if you don't need over-current protection are shown, in Figs.7 & 8. Soldering heavy-duty 1mm² wires to the underside of the board, as shown here, will reduce the resistance of the current-carrying tracks. That will lower the voltage drop between the input and output and allow the PCB to handle more than 5A.**

## Heatsinking

Using, say, IRF1018E and IRF4905 Mosfets, at 4A current draw and 10V or higher, they will dissipate 135mW and 320mW each, respectively. The temperature of a TO-220 package in free air rises by around 70°C/W, so without heatsinking, they will rise to 9°C and 22°C above ambient.

Note that the ambient temperature is the air temperature within the enclosure, which could be significantly higher than room temperature.

The Mosfet on-resistances could be a little higher when using rail voltages significantly below 10V, so for lower operating voltages, pay close attention to the Mosfet temperatures during testing. If they become too warm to touch comfortably, they require heatsinks.

If you know or suspect you'll need heatsinks in advance, it will be easiest to fabricate and mount the transistors onto them before soldering the transistors to the PCB.

You can fashion heatsinks from 3mm aluminium using three separate bars or angles for the three rows of Mosfets. Cut the material to fit comfortably within the component footprint. For a dual rail application, mount two Mosfets per angle with their centre holes spaced 18mm apart. Use insulators and Nylon bushes and/or screws; insulating each Mosfet from its heatsink is the best practice.

You won't necessarily require large heatsinks; it depends on how much power needs to be dissipated. For maximum heat dissipation, bridge the tops of the three aluminium angles with a commercial heatsink.

## Construction

Start by selecting the values of the current sense and bias resistors, and over-voltage threshold zeners, using either the panel or Tables 1 & 2 overleaf.

The Supply Protector is built on a double-sided 96 × 69mm PCB coded 18109241. If you're building the full dual-rail version, fit all the components shown in Fig.3, while if you want to make the single-rail version, fit just the components shown in Fig.5 or Fig.6. Figs.7 & 8 show further variations, which experienced constructors could combine with one of the single-rail variants if desired.

The two adjustment diodes, D4 and D5, are generally not required and can be replaced with wire links. If you need to fine-tune the trip voltage, you can fit diodes instead, as explained earlier.

We suggest constructing in two stages: build and test the reverse-polarity and overvoltage protection sections before adding the remaining components. Roughly speaking, the reverse and overvoltage protection components are below and to the left of the 3300μF electrolytic capacitor in the middle of the board, not including the two 3300μF capacitors (use Fig.7 as a guide).

Pay close attention to the orientation of the transistors, diodes, LEDs and electrolytic capacitors. The two SCRs should face in opposite directions. Start by fitting the lower-profile components like the diodes and resistors that will lie flat on the board (see Table 3), then the capacitors, then the taller components like the Mosfets.

If you want to minimise the voltage drop across the device, or will be using it at high currents, you can solder extra wires to the underside as shown in Fig.4. That should not be necessary for applications up to around 5A per rail, though.

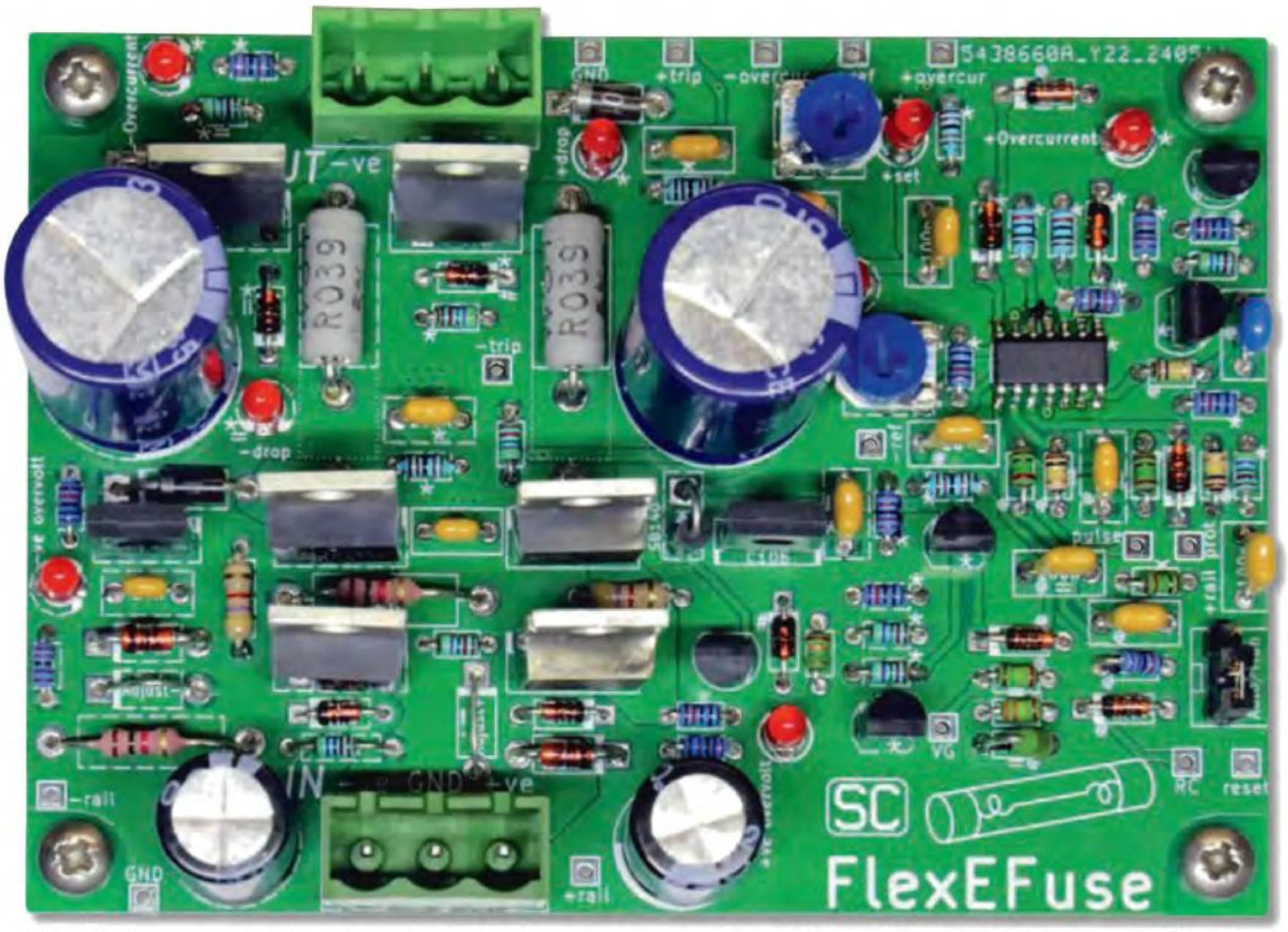
## Initial testing

The easiest way to verify the correct operation of the reverse polarity protection is with a variable power supply. Apply power to the input connector in reverse but starting at 0V. Ramp the voltage slowly up to -1V and monitor the "+rail" and "–rail" test points with a multimeter to verify the absence of any voltage. It is working if the supply reaches -1V and there is no voltage on those test points.

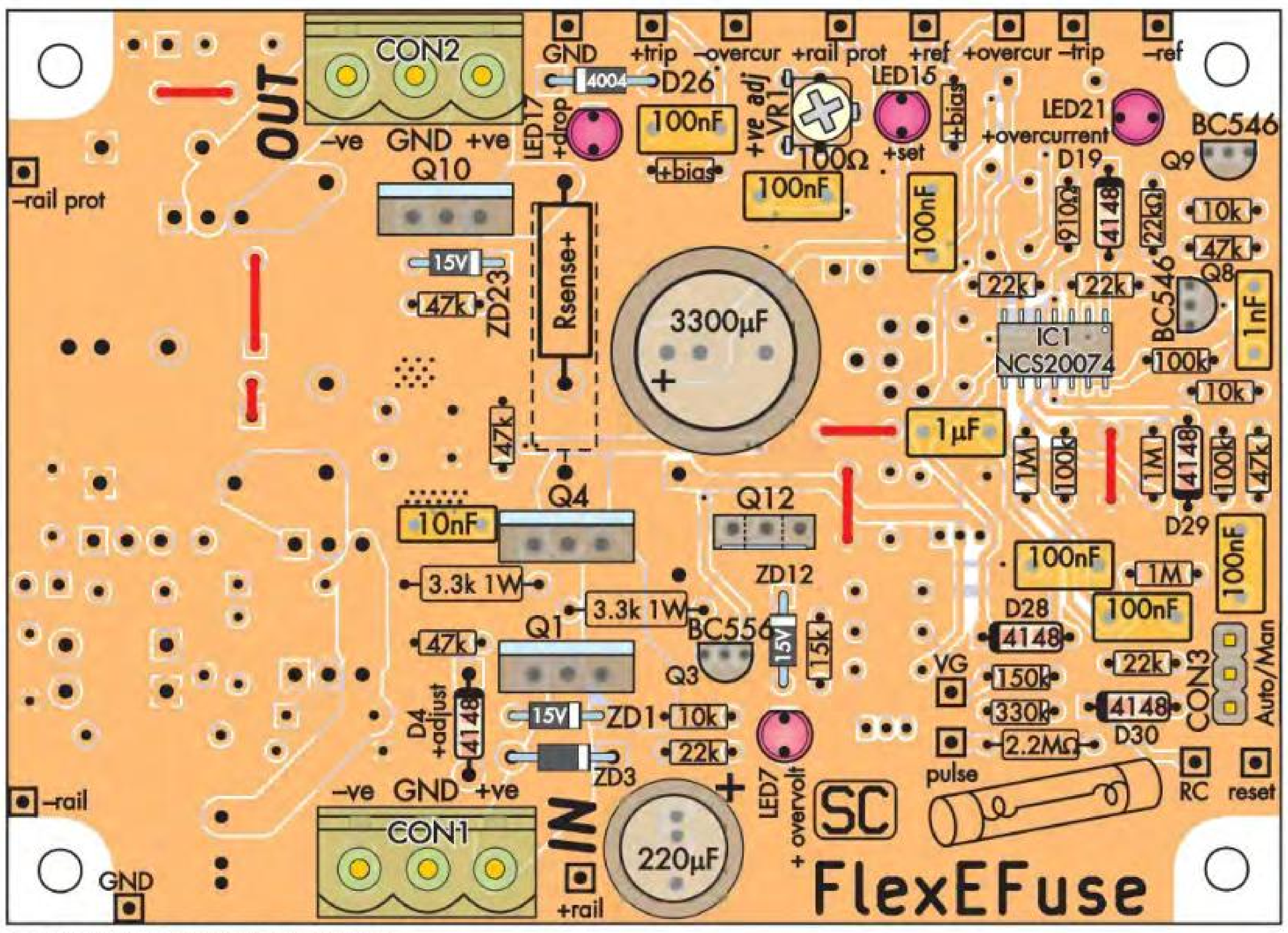
You could use one AA or AAA cell if you don't have a variable power supply.

Note that if you previously applied power in the correct direction, your multimeter may read the residual charge on the 220μF capacitors.

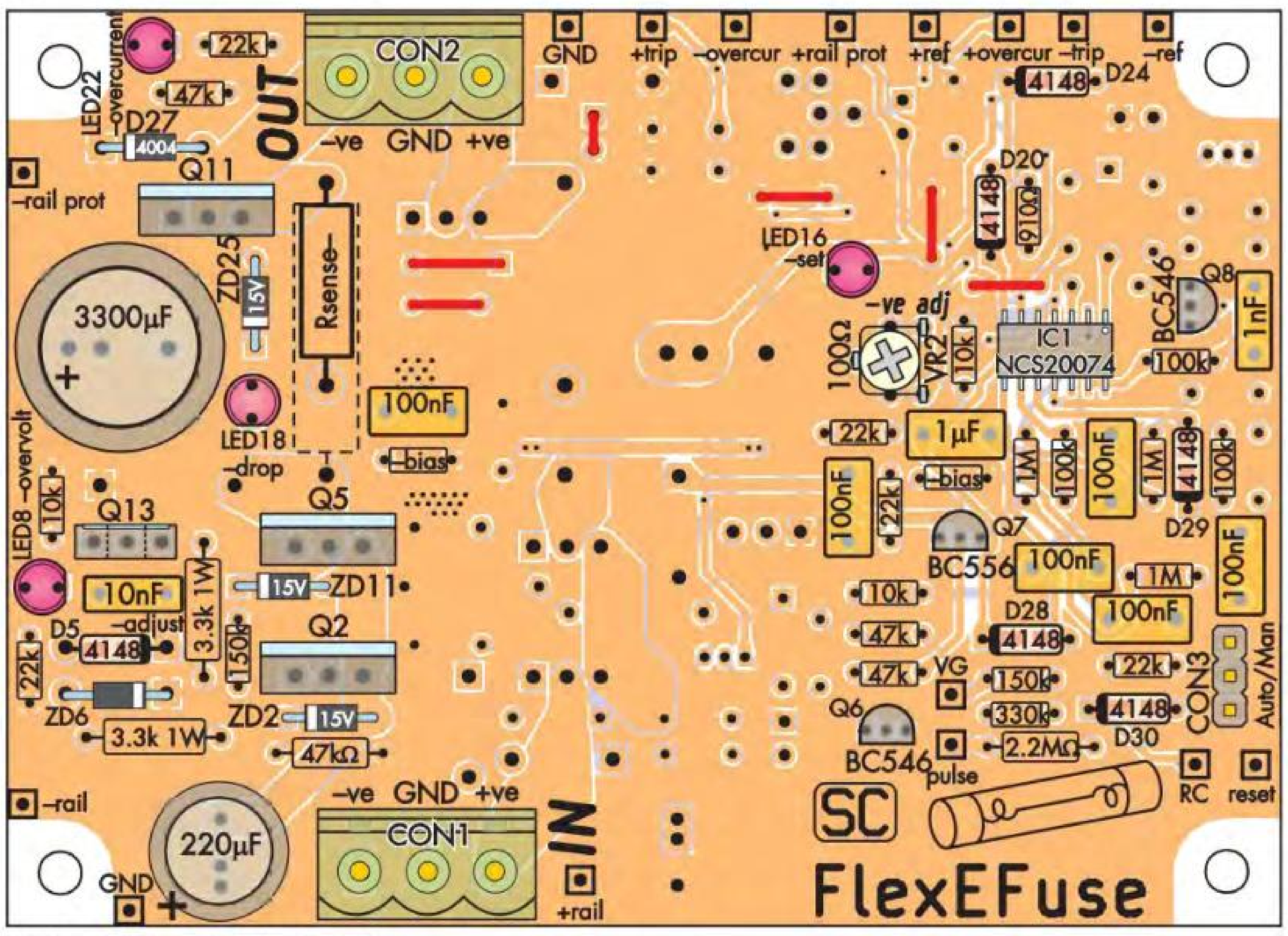
Next, verify the overvoltage protection threshold by switching off the variable power supply and reconnecting the supply with the correct polarity. As you ramp the variable power supply up, by the time it reaches 1V, a voltage will be detectable on both test points.


SC
FlexEFuse
R039
+Overcurrent
-trip
-drop
pulse
GND
+rail
-rail
RC
reset

**The fully assembled Dual Rail Supply Protector PCB with all features available. Note the use of smaller-than-usual resistors to keep it compact.**

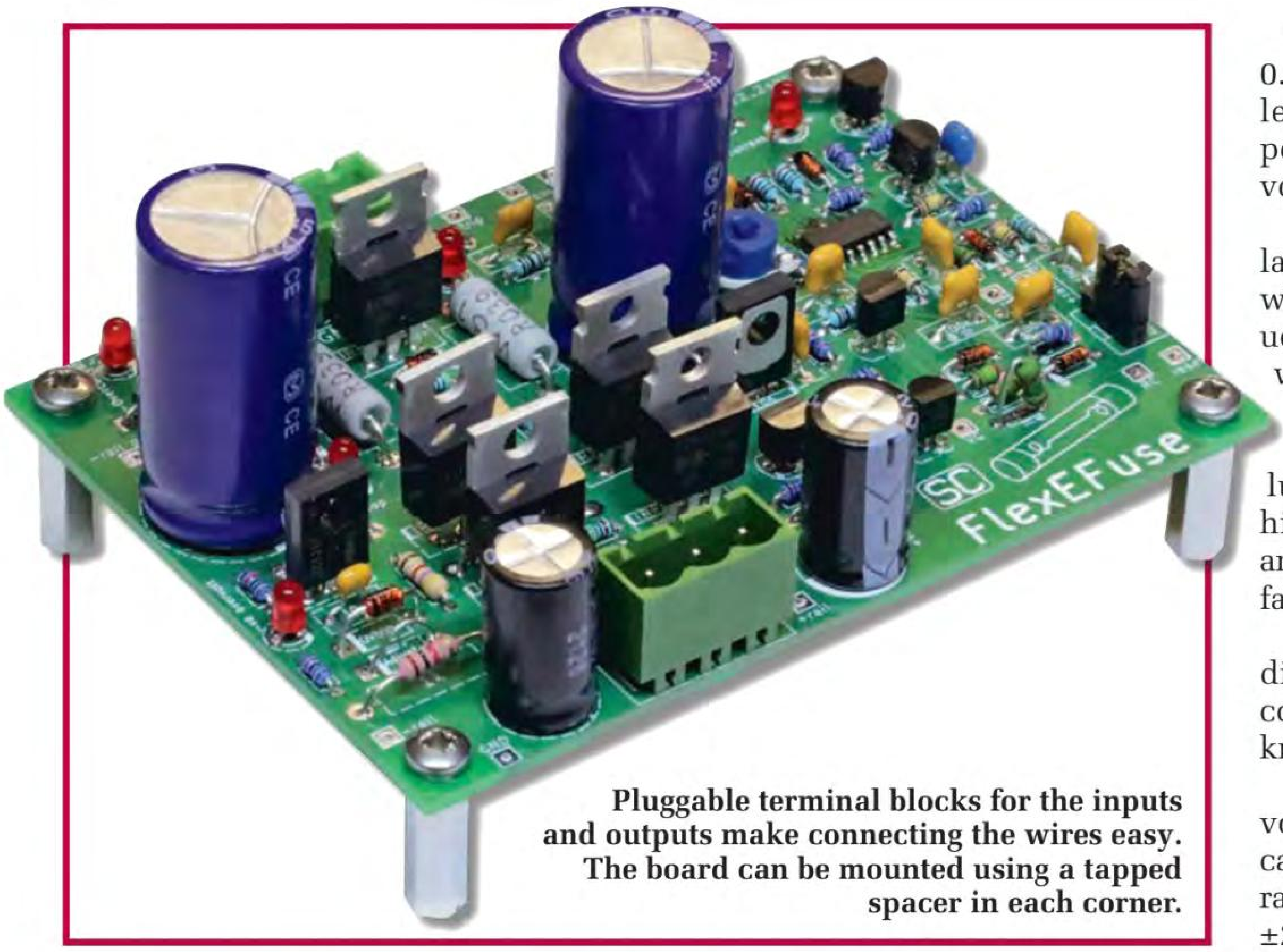

CON2
OUT
-ve GND +ve
GND
+trip
-overcur
+rail prot
+ref
+overcur
-trip
-ref
-rail prot
Q10
D26
4004
LED7 +drop
100nF
+bias
+ve adj
VR1
100Ω
LED15 +set
+bias
LED21 +overcurrent
D19
BC546
Q9
15V
47k
ZD23
Rsense+
3300μF
IC1
NCS20074
22k
10k
47k
100k
1nF
1μF
1M
4148
Q4
Q12
10nF
3.3k 1W
ZD12
BC556
Q1
Q3
15k
D4 +adjust
ZD1
ZD3
D28
D29
D30
150k
330k
2.2MΩ
CON3
Auto/Man
VG
pulse
-ve GND +ve
CON1
IN
+rail
-rail
LED7 +overvolt
220μF
SC
FlexEFuse
RC
reset

POSITIVE RAIL ONLY VERSION

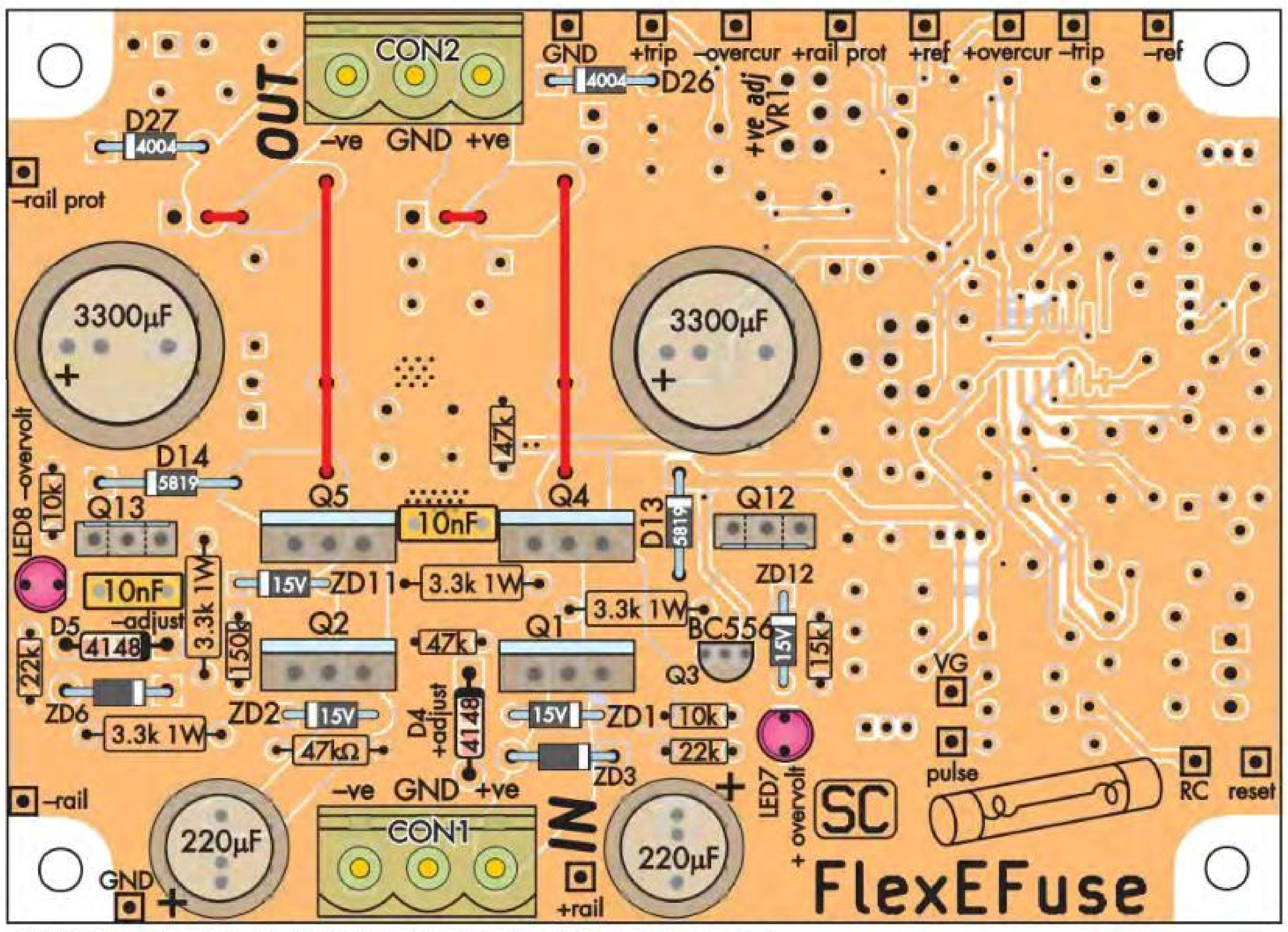

LED22 -overcurrent
22k
47k
D27
4004
CON2
OUT
-ve GND +ve
GND
+trip
-overcur
+rail prot
+ref
+overcur
-trip
-ref
-rail prot
Q11
4148
D24
D20
910Ω
LED16 -set
-ve adj
VR2
100Ω
ZD25
15V
Rsense-
3300μF
IC1
NCS20074
BC546
Q8
1nF
100k
LED18 -drop
100nF
-bias
1μF
1M
22k
LED8 -overvolt
Q13
Q5
10nF
-adjust
D5
ZD11
BC556
Q7
3.3k 1W
150k
Q2
ZD6
ZD2
47kΩ
10k
47k
Q6
BC546
D28
D29
D30
330k
2.2MΩ
CON3
Auto/Man
VG
pulse
-ve GND +ve
CON1
IN
+rail
-rail
220μF
GND
SC
FlexEFuse
RC
reset

NEGATIVE RAIL ONLY VERSION

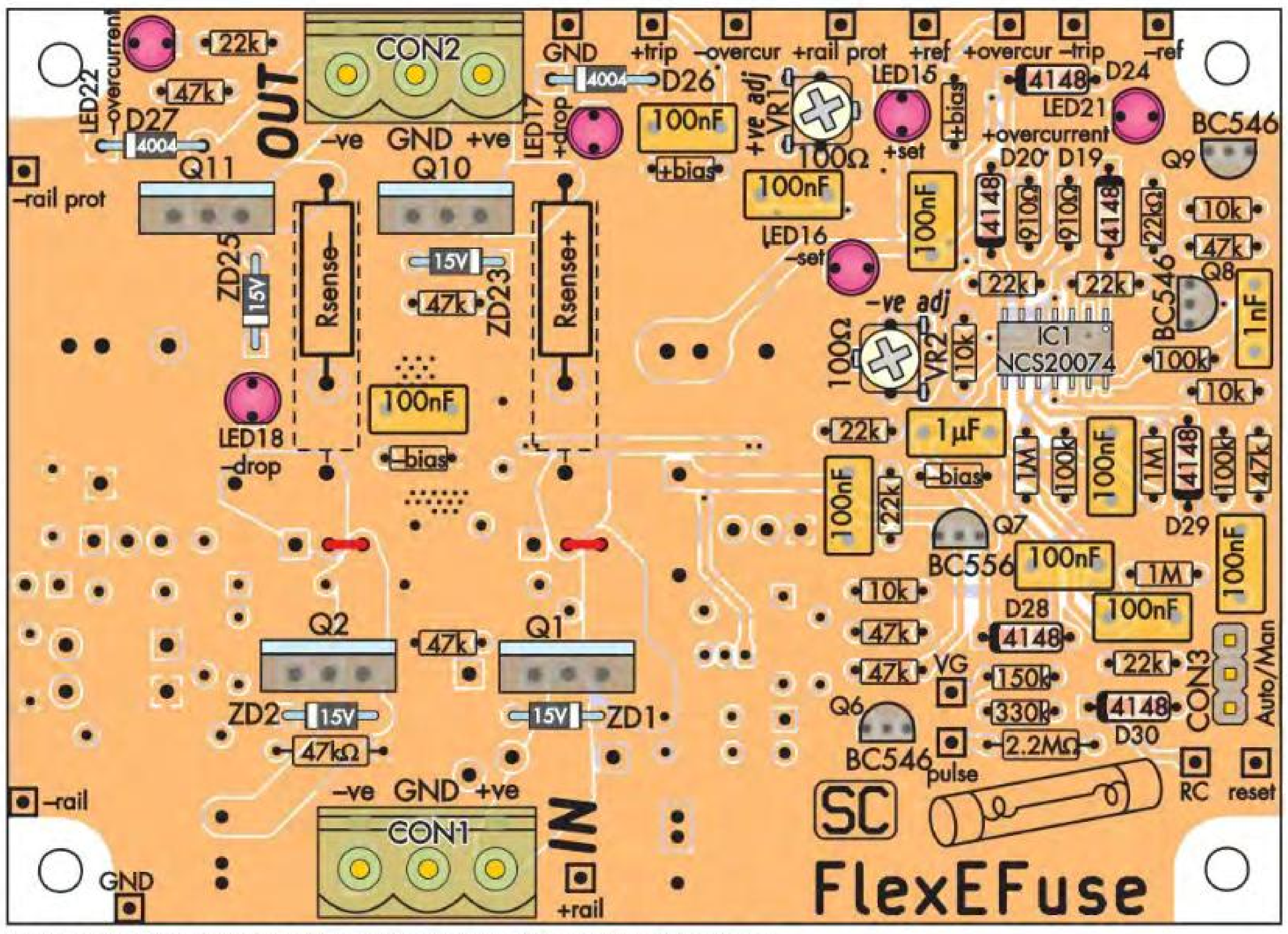
**Figs.5 & 6: these overlay diagrams shows the board fitted with just the components needed for a single rail Supply Protector. You need to add wire links where shown in red. When building any of these versions, watch the orientation of the IC, diodes and Mosfets, as they must all be correct.**


FlexEFuse
SC

**Pluggable terminal blocks for the inputs and outputs make connecting the wires easy. The board can be mounted using a tapped spacer in each corner.**


CON2
OUT
–ve GND +ve
GND +trip –overcur +rail prot +ref +overcur –trip –ref
D26
D27
+ve adj VR1
–rail prot
3300µF
3300µF
D14 5819
Q5
Q4
D13 5819
Q12
LED8 –overvolt
Q13
10nF
10nF
ZD11
3.3k 1W
ZD12
BC556
Q3
D5 4148
–adjust
Q2
47k
Q1
ZD6
3.3k 1W
ZD2
15V
47kΩ
D4 +adjust
ZD1
10k
22k
ZD3
LED7 + overvolt
VG
pulse
RC
reset
–rail
–ve GND +ve
CON1
IN
220µF
220µF
GND
+rail
SC
FlexEFuse

DUAL-RAIL VERSION WITHOUT OVERCURRENT PROTECTION


LED22 –overcurrent
22k
47k
D27
CON2
OUT
–ve GND +ve
GND +trip –overcur +rail prot +ref +overcur –trip –ref
D26
LED17 –drop
100nF
+bias
+ve adj VR1
100Ω
+set
LED15
+bias
4148
D24
+overcurrent
D20 D19
BC546
Q9
LED21
Q11
Q10
–rail prot
100nF
LED16 –set
ZD25
Rsense–
ZD23
Rsense+
15V
47k
–ve adj
100Ω
VR2
IC1 NCS20074
Q8
BC546
1nF
LED18 –drop
100nF
–bias
22k
1µF
–bias
Q7
BC556
D29
100nF
1M
D28
Q2
Q1
ZD2
15V
ZD1
47kΩ
VG
Q6
BC546
150k
330k
2.2MΩ
D30
CON3
Auto/Man
pulse
RC
reset
–rail
–ve GND +ve
CON1
IN
GND
+rail
SC
FlexEFuse

DUAL-RAIL VERSION WITHOUT OVERVOLTAGE PROTECTION

**Figs.7 & 8: these overlays shows which components you can leave off (or link out) if you don't want either the overcurrent or over-voltage protection feature. If building a single-rail version, you will need to refer to Figs.5 & 6 as well, and figure out which components to leave off or link out.**

The readings will initially be around 0.7V below the variable power supply level. Once you reach 3-4V, the test point voltages should rise to the input voltage.

There are two additional test points labelled "+rail prot" and "–rail prot", which you can now monitor. Continue increasing the variable supply towards the protection threshold. As you pass the threshold, each overvoltage protection LED should illuminate and then, at a fractionally higher input voltage, the "+rail prot" and "-rail prot" test points should start falling back to zero.

The actual tripping thresholds may differ from the calculated value due to component tolerances and the zener knees.

Remember not to ramp the input voltage past the ratings of the 220µF capacitors, and be very careful not to ramp your variable supply past 40V (or ±20V) until you are sure the overvoltage protection in both rails is working correctly, or you risk damaging IC1 (if fitted). If either rail's protection hasn't kicked in by 1V beyond the calculated trip point, it's either not working, or the zeners are wrong.

To reset after the overvoltage protection has tripped, return the variable supply to 0V or temporarily disconnect it.

With the overvoltage protection section working, you can finish fitting components to the PCB, starting with IC1; ensure its pin 1 indicator goes at upper right as shown in the overlay diagram. Don't mount the two trimpots yet.

If your board needs cleaning because it's covered in flux residue, submerge it under isopropyl alcohol or methylated spirits and gently rub it with an old toothbrush. Wait for it to dry, then mount and solder the trimpots.

Check that Mosfets are electrically isolated from any heatsink metal.

To verify the overcurrent trip circuit, connect the power rail (or rails) to a variable supply but don't yet connect a load. Adjust the power supply to the same voltage you used to calculate the bias resistances. If either overcurrent trip LED is already illuminated, slowly wind the trimpots clockwise until they extinguish automatically, use the pushbutton reset or short the upper two pins of CON3.

If either of the overcurrent trip LEDs fails to extinguish, there is a problem with the reset or overcurrent circuit.

If you're using auto-reset, check the output of IC1d at the "VG" (virtual ground) test point is near ground level for a symmetrical dual-rail version, or otherwise approximately midway between the power rails.

Assuming both overcurrent LEDs are off and the set LEDs are on, wind each trimpot anti-clockwise until either the trimpot is fully anti-clockwise or the corresponding set LED switches off and the overcurrent LED comes on. When you've reached this point, nudge the trimpot clockwise (and press the reset button if equipped) so both the overcurrent LEDs are again off.

The overcurrent protection circuits will now be armed at a very small current threshold. Use a low-value resistor (say 100Ω or less) between ground and each output rail to verify that the overcurrent protection triggers. The corresponding overcurrent LED should illuminate instantaneously, and the associated set LED should extinguish.

If your overcurrent trip point exceeds the normal operating current by an amp or more, connect a dummy load that will draw current just below the overcurrent tripping point and adjust VR1 and VR2 so the load remains turned on. Otherwise, use your intended load to set the overcurrent trip point.

Finally, run the device for ten minutes while monitoring the temperature of the Mosfets. If the Mosfets become too warm to touch comfortably, turn off the power and fit heatsinks before using it. Wire it up, and you can sit back and relax, knowing your load device is protected!

## Parts List – DC Supply Protectors (all features)

1 double-sided PCB coded 18109241, 96 × 69mm
2 100Ω miniature top-adjust trimpots (VR1, VR2)
2 3-way 5.08mm pitch pluggable terminal blocks (CON1, CON2)
**[Farnell 3797901 + 2452499 or DigiKey 5EHDVC-03P + 5ESDV-03P]**
1 3-way pin header and jumper shunt (CON3)
1 NO momentary pushbutton (**optional**)
4 M3 × 6mm panhead machine screws and matching spacers

**Semiconductors**
1 NCS20074 quad rail-to-rail output op amp, SOIC-14 (IC1)
3 high-current P-channel Mosfets, TO-220 (Q1, Q4, Q10) ★
3 high-current N-channel Mosfets, TO-220 (Q2, Q5, Q11) ★
2 BC556 45V 100mA PNP transistors, TO-92 (Q3, Q7)
3 BC546 45V 100mA NPN transistors, TO-92 (Q6, Q8, Q9)
2 C106 SCRs, TO-126 (Q12, Q13)
8 high-brightness 3mm LEDs (LED7-8, LED15-18, LED21-22)
**[Vishay TLLK4401]**
6 15V 1A zener diodes, DO-41 (ZD1, ZD2, ZD11, ZD12, ZD23, ZD25)
2 1A zener diodes, DO-41, values to suit application – see Table 1 (ZD3, ZD6)
2 1N5819 or SB140 40V 1A schottky diodes (D13 & D14)
6 1N4148 75V 200mA diodes, DO-35 (D19-D20, D24, D28-D30)
2 1N4004 400V 1A diodes, DO-41 (D26, D27)
2 extra diodes to fine-tune over-voltage thresholds (D4, D5; **optional, see text**)
★ *suitable types include IRF4905 (up to ±55V & 4A), IPP80P03P4L-07 (±30V & 7A) and SUP90P06-09L-E3 (±60V & 7A)*
★ *suitable types include IRF1018E (7A), CSD18534KCS (7A), DIT050N06 (4A), STP60NF06 (5A) & IPP80N06S4L (8A; all can handle up to ±60V, ratings are without heatsinks and are only a guide)*

**Capacitors**
2 3300µF 50V electrolytic (5mm or 7.5mm lead pitch, diameter ≤ 18mm)
2 220µF 63V low-ESR electrolytic (3.5mm or 5mm lead pitch)
1 1µF 50V ceramic or MKT
9 100nF 50V ceramic or MKT
2 10nF 50V ceramic or MKT
1 1nF 50V ceramic or MKT

**Resistors** *(all ⅛W 5% miniature axial unless noted)* ◆
1 2.2MΩ
3 1MΩ
1 330kΩ
2 150kΩ
3 100kΩ
9 47kΩ
9 22kΩ
1 15kΩ
6 10kΩ
4 3.3kΩ 1W 5%
2 910Ω
4 $R_{BIAS}$ resistors – see Table 1
2 $R_{SENSE}$ resistors – see Table 2
◆ *regular ¼W resistors can be used but they won't sit flat on the PCB*

### Table 1 – zener diode values

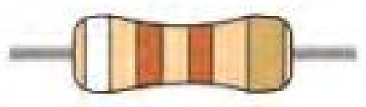
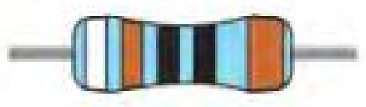
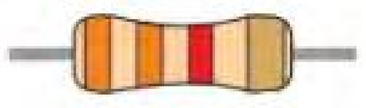
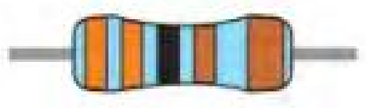
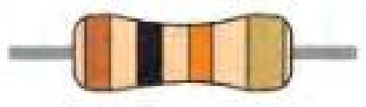
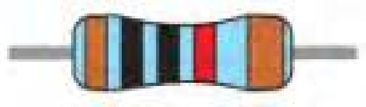
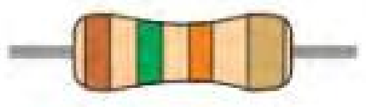
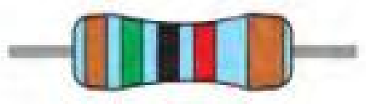
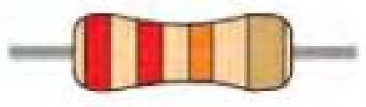
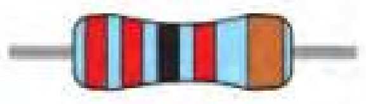
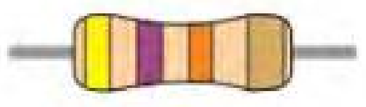
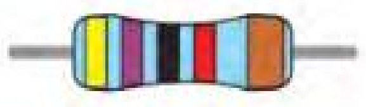
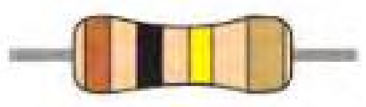
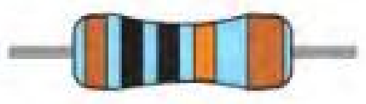
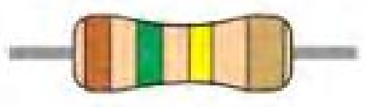
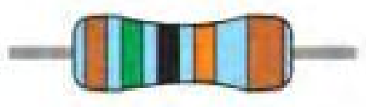
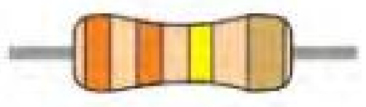
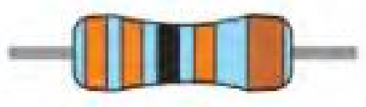
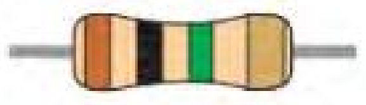
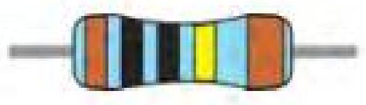
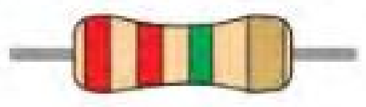
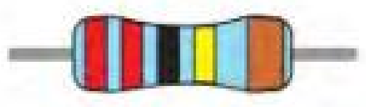
| Trip | ZD3/ZD6 | Bias resistors |
|---|---|---|
| ~5V | 5.1V | 3.0kΩ |
| ~5.5V | 5.6V | 3.3kΩ |
| ~7.25V | 6.8V | 5.1kΩ |
| ~10.3V | 10V | 8.2kΩ |
| ~13V | 13V | 11kΩ |
| ~15.1V | 15V | 13kΩ |
| ~18V | 18V | 16kΩ |
| ~20V | 20V | 18kΩ |
| ~23.8V | 24V | 22kΩ |
| ~29.5V | 30V | 27kΩ |
| ~38V | 39V | 36kΩ |

### Table 2 – current sense resistors

| Adjustment range | Sense resistor |
|---|---|
| 0-1A | 100mΩ 1/8W |
| 0-2A | 50mΩ 1/4W |
| 0-3A | 33mΩ 1/4W |
| 0-4A | 25mΩ 1/2W |
| 0-6A | 16mΩ 1/2W |
| 0-8A | 12mΩ 2/3W |
| 0-10A | 10mΩ 1W |

**Tables 1, 2 and the panel on the next page are used to determine the best values of various components to suit your needs.**

### Table 3 – resistor colour codes

| Value | 4-band code | 5-band code |
|---|---|---|
| 910Ω |  |  |
| 3.3kΩ |  |  |
| 10kΩ |  |  |
| 15kΩ |  |  |
| 22kΩ |  |  |
| 47kΩ |  |  |
| 100kΩ |  |  |
| 150kΩ |  |  |
| 330kΩ |  |  |
| 1.0MΩ |  |  |
| 2.2MΩ |  |  |

## Calculating component values

Several component values should be selected to suit your application as follows.

**Overvoltage trip point**

First, you must determine the highest voltage that's safe to apply to the load. If unsure, measure the output of the existing power supply and add a safety margin.

Zener diodes ZD3 ('+OVtrip') and ZD6 (-OVtrip') set the overvoltage trip point for each rail in combination with the 3.3kΩ resistors. The SCRs will trip when their trigger input reaches approximately 0.6V. Allowing for a voltage drop of about 100mV across the resistors, the required zener voltage is (Trip – 0.7V) × 1.05.

As mentioned earlier, low-voltage zeners may trigger at lower voltages than expected. Also, typical zeners diodes have 5% tolerances.

In the middle of the voltage range (eg, around ±15V), you can generally get away with a zener diode that has a voltage rating close to the desired trip point, as the 0.7V and 5% factors cancel out.

Because the expected overvoltage trip point lies within a range, and zeners are only available in certain preferred values, you may need to use adjustment diodes if you require high precision. Adding a schottky diode for D4 or D5 (like a BAT85, SB140 or 1N5819) will increase that rail's trip point by around 0.3V, while adding a silicon diode (like a 1N4148 or 1N4004) will increase it by around 0.6V.

Don't use zeners below 4.3V or above 19V (for a dual-rail configuration) or 39V (for single-rail operation). You can use different values for the two zeners for asymmetric applications.

Ensure that the 3300µF output capacitors have voltage ratings above the trip points. For example, if you have ±18.1V overvoltage protection thresholds, select 25V capacitors.

Because the 220µF capacitors after Mosfets Q1 & Q2 are on the unprotected side of the overvoltage protection circuit, they will experience any overvoltage, so their voltage ratings should exceed the highest expected input voltage. We recommend using 50V or 63V rated capacitors there, although you might get away with 35V caps in some cases.

**Overcurrent trip point**

The '+sense' and '-sense' resistors are used to monitor the current in each rail. The overcurrent trip is calculated for a sense resistor voltage drop of about 50mV, although the trimpots let you set it up to 100mV.

Use Ohm's law, R = V/I, and the power formula, P = VI, to calculate the required resistances and power ratings. Let's use 2A as an example. For a 50mV drop, the formulas give R = 25mΩ (0.05V ÷ 2A). If you can't find a resistor with the calculated value, round the resistance to the closest available value. A 0.022Ω, 0.025Ω or 0.033Ω resistor would be suitable in this case.

We calculate the power at 100mV as we don't want the resistor to overheat if the trimpot is set to maximum, so P = 200mW (0.1V × 2A). Ideally, the resistor should have close to twice the power rating (to account for elevated ambient temperatures etc), so in this case, use a ½W or 0.6W resistor.

If your application has asymmetric current requirements, you can choose different values for the two resistors.

**Bias resistances**

Four resistors are labelled '+bias' or '-bias'. The bias resistors are selected so that about 1mA flows through them when the supply is at its nominal (not overvoltage trip) level.

The series LEDs have a forward voltage drop of around 1.8-2V, so consider that when calculating the resistor values. The exact drop doesn't matter as long as the four LEDs (LED15-LED18) are the same type, so the voltage drops are similar. Red, orange or yellow LEDs with a forward voltage drop below 2.3V will work.

You can measure the LED's forward-biased drop using a digital multimeter's diode testing function.

Say the nominal power supply is ±12V and you have red LEDs with a 1.6V forward voltage. The required resistance will be R = (12V – 1.6V) ÷ 0.001A = 10.4kΩ. Choose the nearest available resistance, 10kΩ in this case.

If you have an application with asymmetric voltage rails, the '+bias' and '-bias' resistances may differ.

NEW!

5-year collections

2019-2023

All 60 issues from Jan 2019 to Dec 2023 for just £49.95

PDF files ready for immediate download

2018-2022

All 60 issues from Jan 2018 to Dec 2022 for just £49.95

PDF files ready for immediate download

2017-2021

All 60 issues from Jan 2017 to Dec 2021 for just £49.95

PDF files ready for immediate download

2016-2020

All 60 issues from Jan 2016 to Dec 2020 for just £44.95

PDF files ready for immediate download

See page 69 for further details and other great back-issue offers.

Purchase and download at:
www.electronpublishing.com

# DOWNLOAD YOUR PE/EPE BACK ISSUES

## COMPLETE-YEAR collections – all 12 issues for each year!

**From just £5.95 each**

**VOL 28** – January to December 1999 – **£5.95**
**VOL 29** – January to December 2000 – **£6.95**
**VOL 30** – January to December 2001 – **£7.95**
**VOL 31** – January to December 2002 – **£7.95**
**VOL 32** – January to December 2003 – **£7.95**
**VOL 33** – January to December 2004 – **£7.95**
**VOL 34** – January to December 2005 – **£8.95**
**VOL 35** – January to December 2006 – **£8.95**
**VOL 36** – January to December 2007 – **£8.95**
**VOL 37** – January to December 2008 – **£8.95**
**VOL 38** – January to December 2009 – **£10.95**
**VOL 39** – January to December 2010 – **£11.95**
**VOL 40** – January to December 2011 – **£12.95**
**VOL 41** – January to December 2012 – **£14.95**
**VOL 42** – January to December 2013 – **£15.95**
**VOL 43** – January to December 2014 – **£16.95**
**VOL 44** – January to December 2015 – **£17.95**
**VOL 45** – January to December 2016 – **£19.95**
**VOL 46** – January to December 2017 – **£22.95**
**VOL 47** – January to December 2018 – **£27.95**
**VOL 48** – January to December 2019 – **£27.95**
**VOL 49** – January to December 2020 – **£29.95**
**VOL 50** – January to December 2021 – **£32.95**
**VOL 51** – January to December 2022 – **£34.95**
**VOL 52** – January to December 2023 – **£34.95 – NEW!**
**VOL 53** – January to December 2024 – **£34.95 – NEW!**

## FIVE-YEAR collections – 60 issues!

**From just £14.95 each**

**PE2832: Jan 99 to Dec 03 – £14.95**
**PE2933: Jan 00 to Dec 04 – £14.95**
**PE3034: Jan 01 to Dec 05 – £14.95**
**PE3135: Jan 02 to Dec 06 – £14.95**
**PE3236: Jan 03 to Dec 07 – £14.95**
**PE3337: Jan 04 to Dec 08 – £14.95**
**PE3438: Jan 05 to Dec 09 – £19.95**
**PE3539: Jan 06 to Dec 10 – £25.95**
**PE3640: Jan 07 to Dec 11 – £25.95**
**PE3741: Jan 08 to Dec 12 – £25.95**
**PE3842: Jan 09 to Dec 13 – £33.95**
**PE3943: Jan 10 to Dec 14 – £33.95**
**PE4044: Jan 11 to Dec 15 – £33.95**
**PE4145: Jan 12 to Dec 16 – £35.95**
**PE4246: Jan 13 to Dec 17 – £37.95**
**PE4347: Jan 14 to Dec 18 – £38.95**
**PE4448: Jan 15 to Dec 19 – £39.95**
**PE4549: Jan 16 to Dec 20 – £42.95**
**PE4650: Jan 17 to Dec 21 – £44.95**
**PE4751: Jan 18 to Dec 22 – £44.95 – NEW!**
**PE4852: Jan 19 to Dec 23 – £44.95 – NEW!**

## 20 YEARS OF PE!

AMAZING BUNDLE OFFER
Every issue from 1999 – 2018

240 magazines for just

**£69.95**

**PE2832: Jan 99 to Dec 03**
**PLUS!**
**PE3337: Jan 04 to Dec 08**
**PLUS!**
**PE3842: Jan 09 to Dec 13**
**PLUS!**
**PE3447: Jan 14 to Dec 18**

**Download your collections at: www.electronpublishing.com**

# AUDIO OUT

By Jake Rothman

## Discrete discoveries and a cascode amplifier PCB design

**I'm waiting for a delivery from JLCPCB** near Hong Kong. When the PCBs (designed by Mike Grindle) arrive, we'll have an efficient way of building the cascode-input power amplifier and its unusual power supply, described in the July issue. Meanwhile, I can report results of some further investigations into the design.

### Baxandall balanced input

Since my cascode amplifier circuit can be configured as inverting, it seemed logical to try adding an input balancing transformer using the Baxandall configuration described last month. This led to some interesting effects.

It has to be admitted this is a risky approach, since we are expecting the limited negative feedback to linearise three things at once: the output stage, output capacitor and input transformer.

In analog design, it's desirable to linearise one thing at a time with separate loops, hence the profusion of op amps used in modern designs. Trying to make negative feedback (NFB) do too much can lead to complex interactions and odd frequency effects.

In the days of valves and expensive transistors, though, this approach was normal, with NFB being taken around multiple capacitors and transformers. To get this to work without the circuit bursting into low-frequency oscillation ('motorboating') involves a lot of component value tweaking.

This is the sort of design challenge I enjoy, because you get more out of less. It's more creative than stringing together standard circuits and is a privilege only afforded to the home constructor or sole trader today.

### Bumps in the frequency domain

As expected, LF oscillation occurred at around 8Hz when adding the modem transformer to the input (as shown in Fig.4(a) in the August 2025 issue). This was soon tamed by changing the low-frequency (LF) step network resistor, R3, to 11kΩ. All that remained was a tolerable peak of 1.5dB at 13Hz.

This sort of tweaking is easy to do with a signal generator and oscilloscope; 'posh' analysers such as the Audio Precision don't go down low enough.

### Distressing distortion?

Of course, the AP is very useful for checking the distortion, which is where some intractable problems were revealed. The Baxandall balanced input depends on a low-impedance virtual ground on the inverting input to work.

Unfortunately, the input DC blocking capacitor (C1) and the LF step network raise the impedance at low frequencies, which causes the distortion to increase rapidly below 200Hz, as shown in Fig.1. It was still much better than connecting the transformer in the normal way, though.

Subjectively, the effect was enhancing rather than destructive, especially on electro jazz pop, such as Level 42 ("you old codger", I bet you're thinking!).

### Switch off or carry on?

The utility of a fully floating (Earth-isolated) transformer input cannot be understated. I found the switch-mode power supply buzz and monitor noise from my desktop computer to be completely eliminated when being used as a sound source.

With a stereo amp using a single power supply, the left and right amplifier signal Earths have to be joined together at the input jack. This causes an Earth loop, which invites hum. Putting ground lift resistors in can fix this, but distortion then increases. Using transformers breaks the loop.

The amp also excelled when used as mid-range and tweeter amps in an active speaker system with a common power supply for all three amplifiers. Floating differential inputs greatly simplify Earth loop problems. The LF distortion was a long way from the required bandwidth, so it had no effect.

### Building bridges

Until the boards arrive, I can't test the transformer bridge-mode version of the amp, so it's time for another diversion and back to Blumlein. Normally, when making a bridging driver, a circuit that gives two anti-phase outputs is used, such as two op amps, wired as an inverter and buffer. Very boring; I've already deployed 100 NE5532s in the last year.

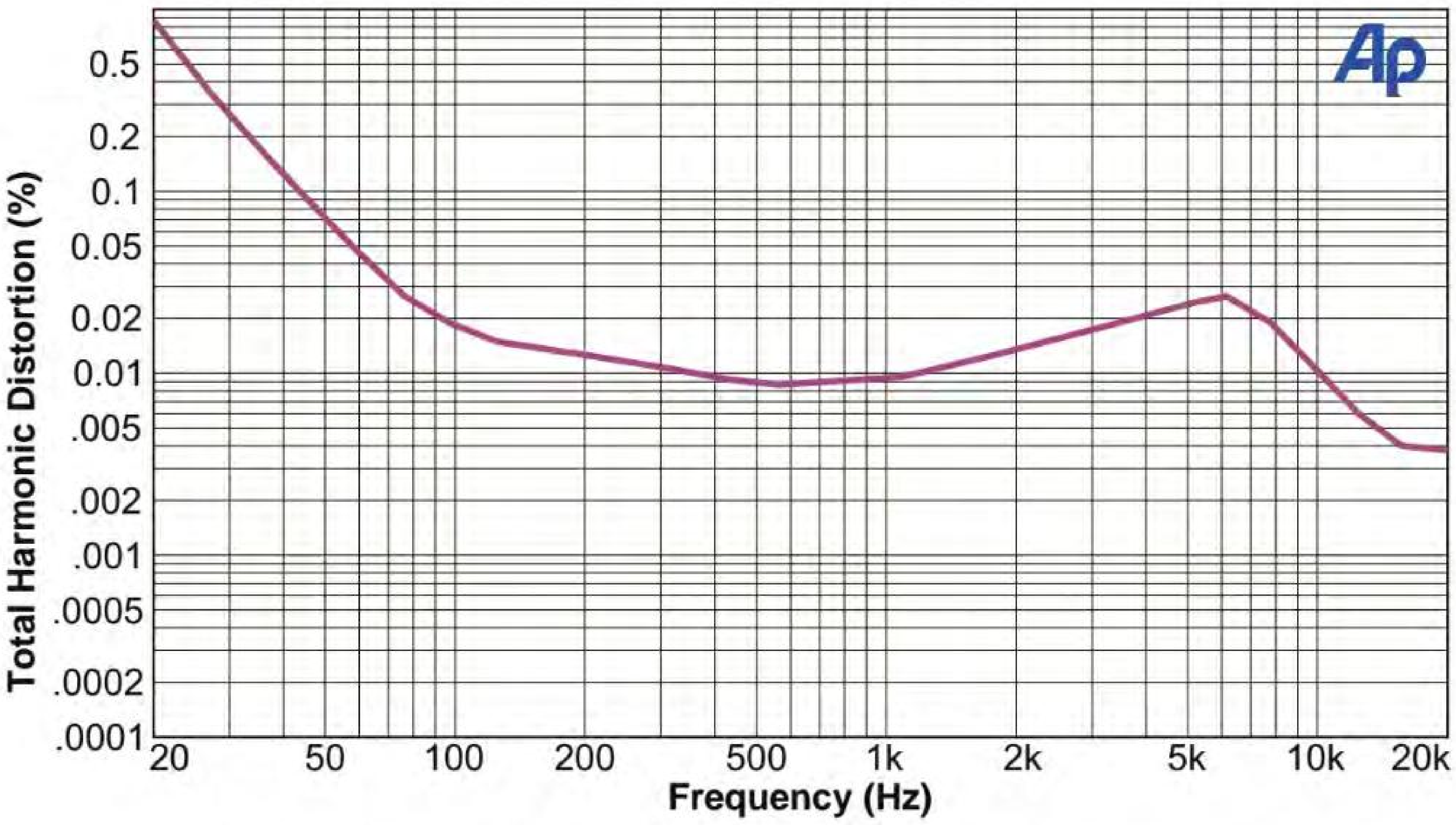


Fig.1: the distortion curve with a balancing transformer added to the cascode power amp. Note the rise in LF distortion. The output is 15V peak-to-peak into 8Ω.

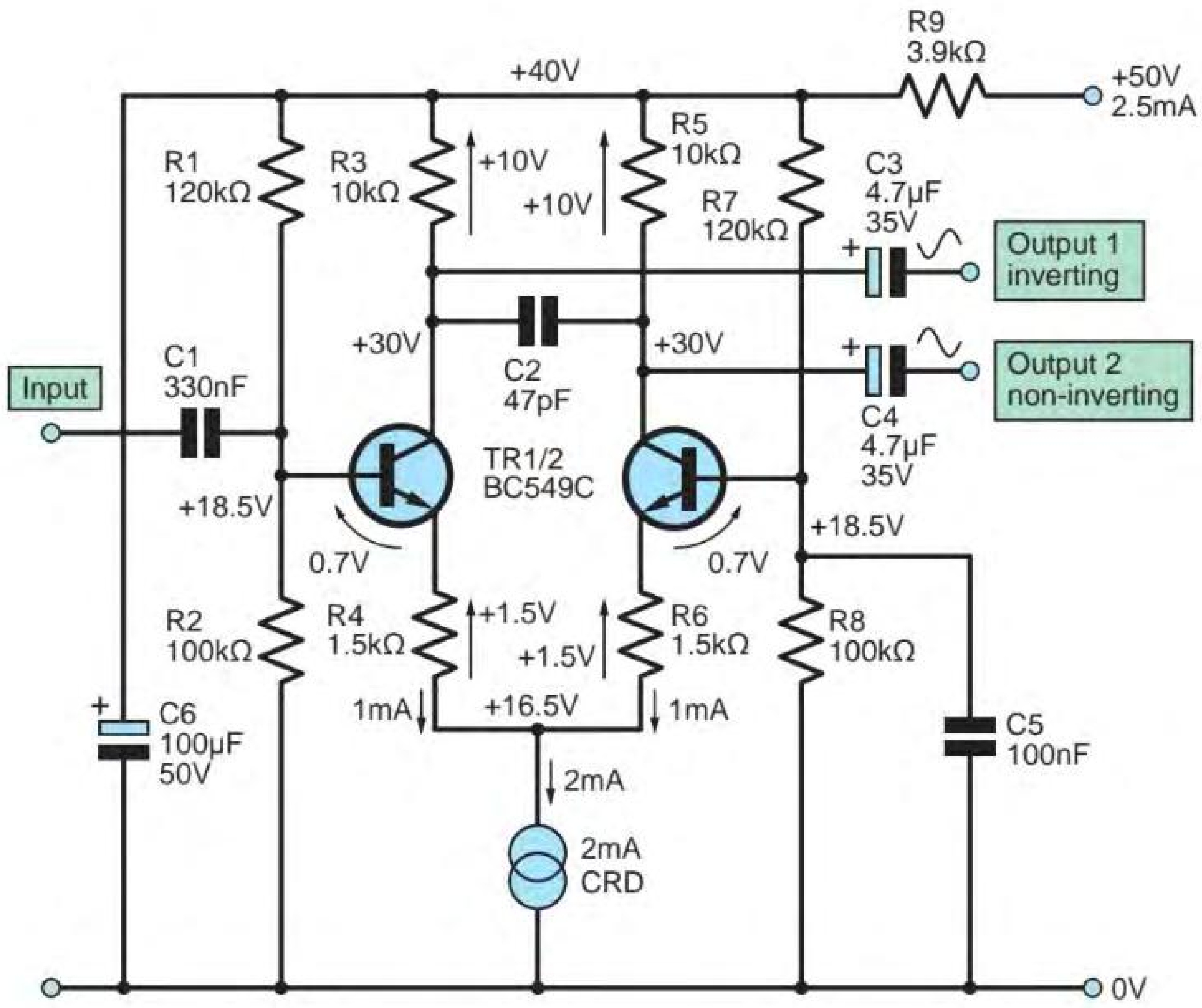


Fig.2: the LTP phase-splitter seems to have been forgotten. It can have single-ended or balanced inputs.

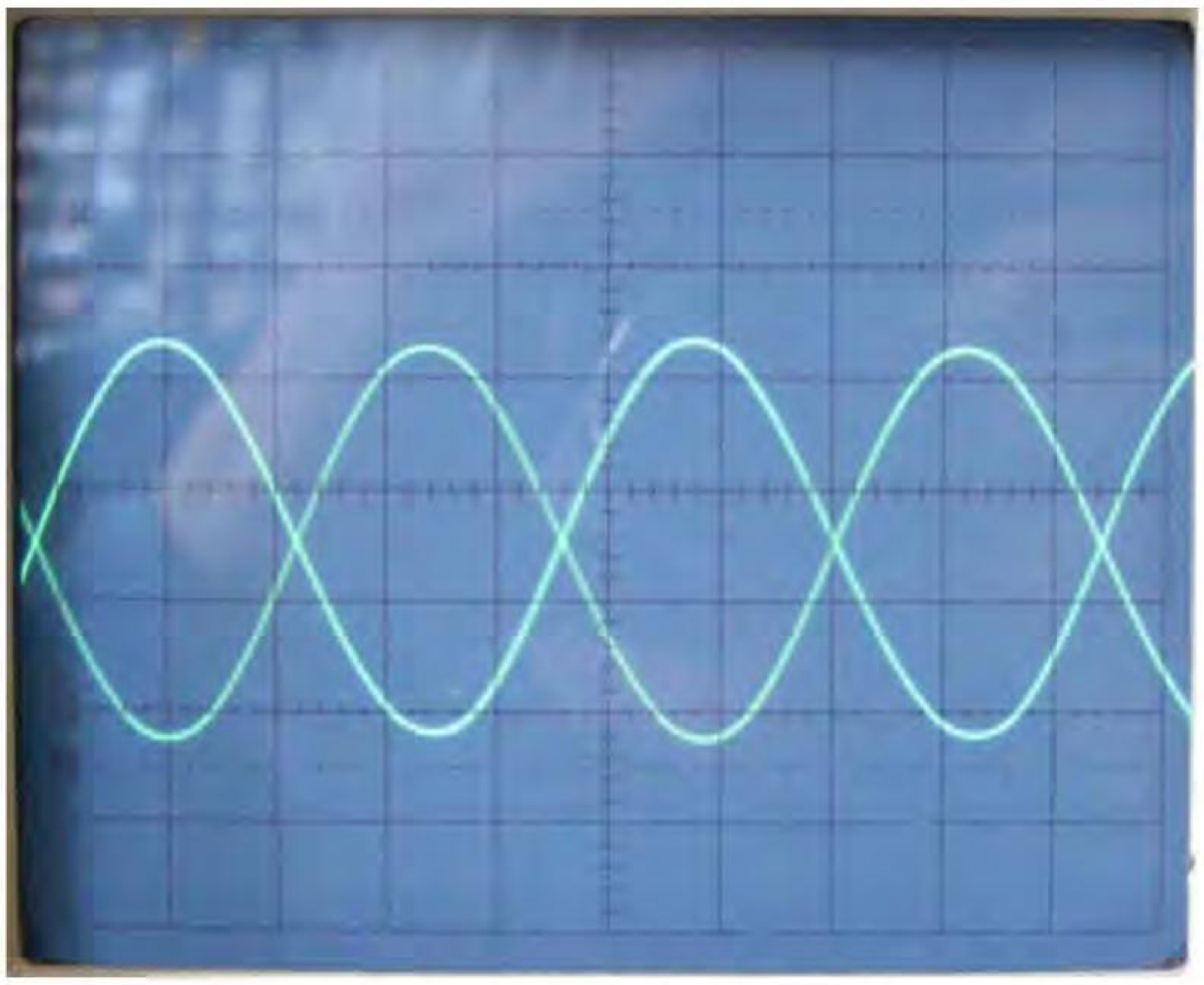


Photo 1: the outputs of a phase splitter are the same amplitude.

It's an ideal application for Blumlein's major legacy, the long-tailed pair.

A circuit that gives two anti-phase outputs from a single input used to be commonly known as a phase splitter. These were an important stage in valve amplifiers, where they drove the two output valves in push-pull. The classic circuit still used today is the Mullard 510, which used a long-tailed pair built around a double-triode ECC83 valve.

I thought it would be useful to design a transistor version to add to my discrete circuit armoury. Older amplifiers used a single valve or transistor to accomplish phase splitting, called a concertina phase splitter.

This had the disadvantage of the load on one output affecting the other output. The LTP phase splitter doesn't suffer from this defect.

## LTP phase splitter

The circuit is shown in Fig.2. The first version I built set the tail current with an 8.2kΩ resistor, textbook style, which was a bad idea; not all the signal was transferred to the second transistor, some was lost in the resistor. This resulted in the second output being 20% lower than the first. The whole point of a phase-splitter is equal balance.

The horrid sound and inefficiency of one amplifier in a bridge clipping, while the other amplifier is clean, defeats the whole point of the system. If the tail resistor is replaced by a 2mA current regulating diode (CRD), as shown in Photo 1, the imbalance is fixed. Alternatively, a constant current sink built around a JFET or bipolar transistor could be used.

The op amp phase-splitter has the advantage of being able to drive low impedances, down to 1kΩ or so, with insignificant distortion. This phase-splitter has a high output impedance, defined by the collector resistors R3 and R5. Replacing these resistors with a current mirror would double the gain and output current capability for the same supply current.

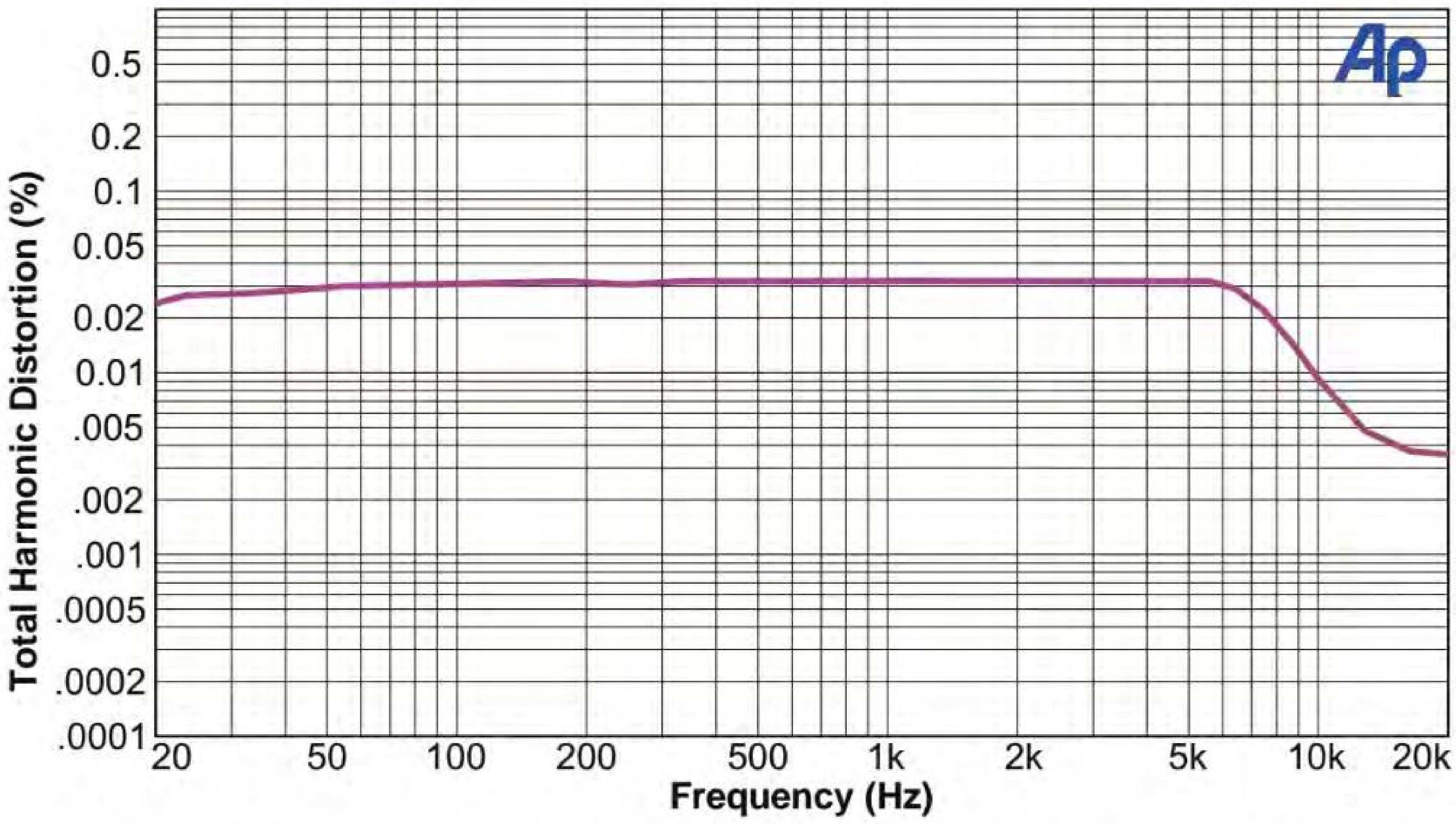


Fig.3: the LTP phase splitter distortion with a 20V peak-to-peak output (unloaded). It's quite high, but is essentially innocuous third harmonic at a consistent level.

I measured the unloaded distortion with an 8V peak-to-peak signal at 0.03%, as shown in Fig.3. It was mainly third-harmonic, arising from the start of the symmetrical soft clip curvature.

Such distortion is very innocuous and similar to that produced by speaker excursion limiting and tape saturation.

Interestingly, the distortion did not increase with 600Ω loading, although the output voltage dropped.

One thing I've noticed with differential operation is that the distortion of tantalum coupling capacitors cancels out because it is equivalent to operating two capacitors wired back-to-back. The distortion is further minimised in single-rail designs such as this, because the capacitors are polarised. So there is no characteristic bass tip-up in the distortion curve.

The gain of the circuit for each output is 3.5 times, defined by the emitter resistors along with the collector loading. With differential outputs, there is a further gain doubling because one output goes up as the other goes down. The frequency response measures -1dB at 20Hz and 20kHz.

The output rolls off around 300kHz, with output 2 dropping first because of the extra losses passing through an extra transistor. The capacitance of the constant current sink also has an effect.

Maybe the long-forgotten Baxandall bootstrapped complementary constant current circuit would work well here, but that's another investigation. Alternatively, I could design a cascode phase-splitter that needs no current sink at all. A practical, breadboarded

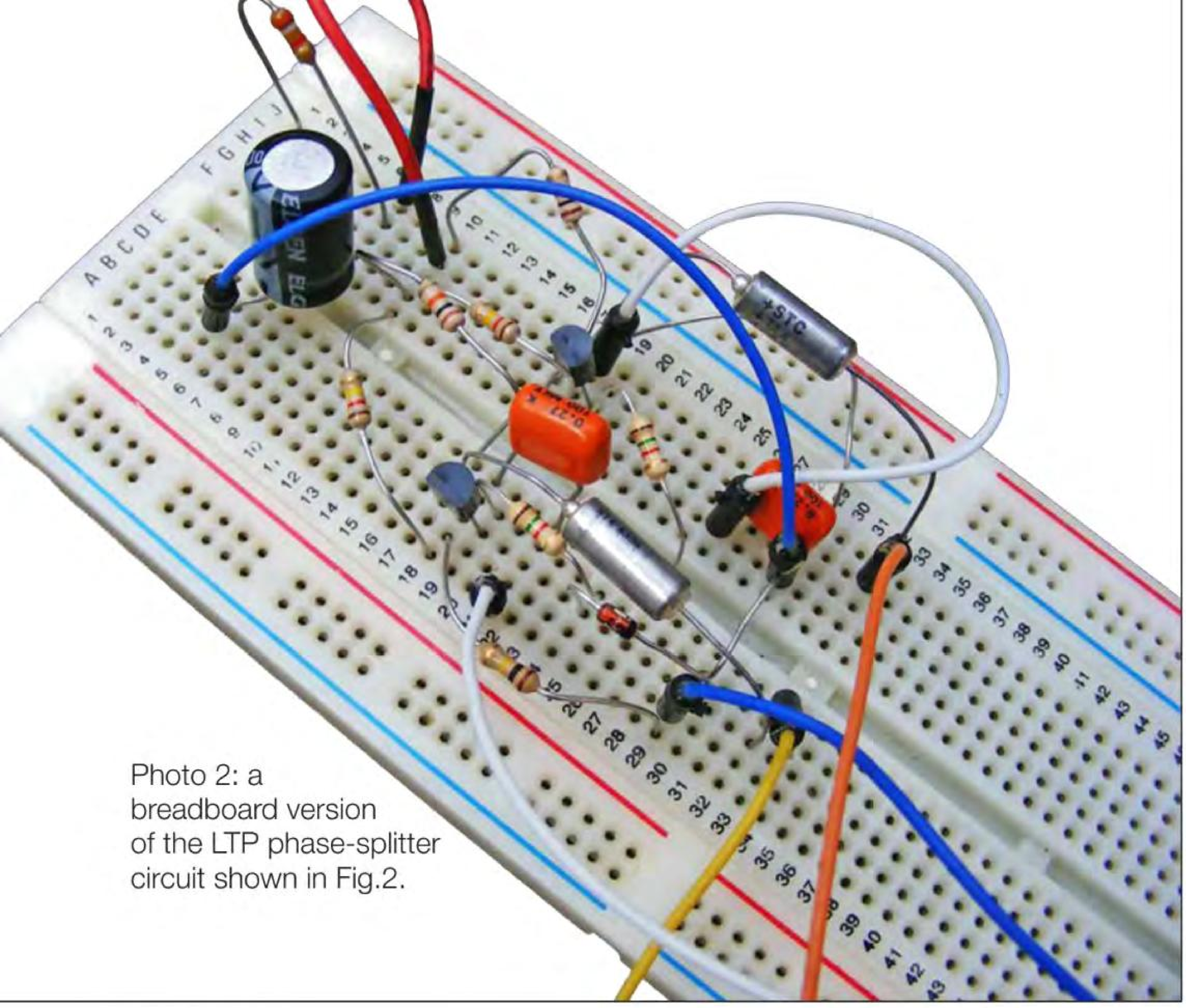


Photo 2: a breadboard version of the LTP phase-splitter circuit shown in Fig.2.

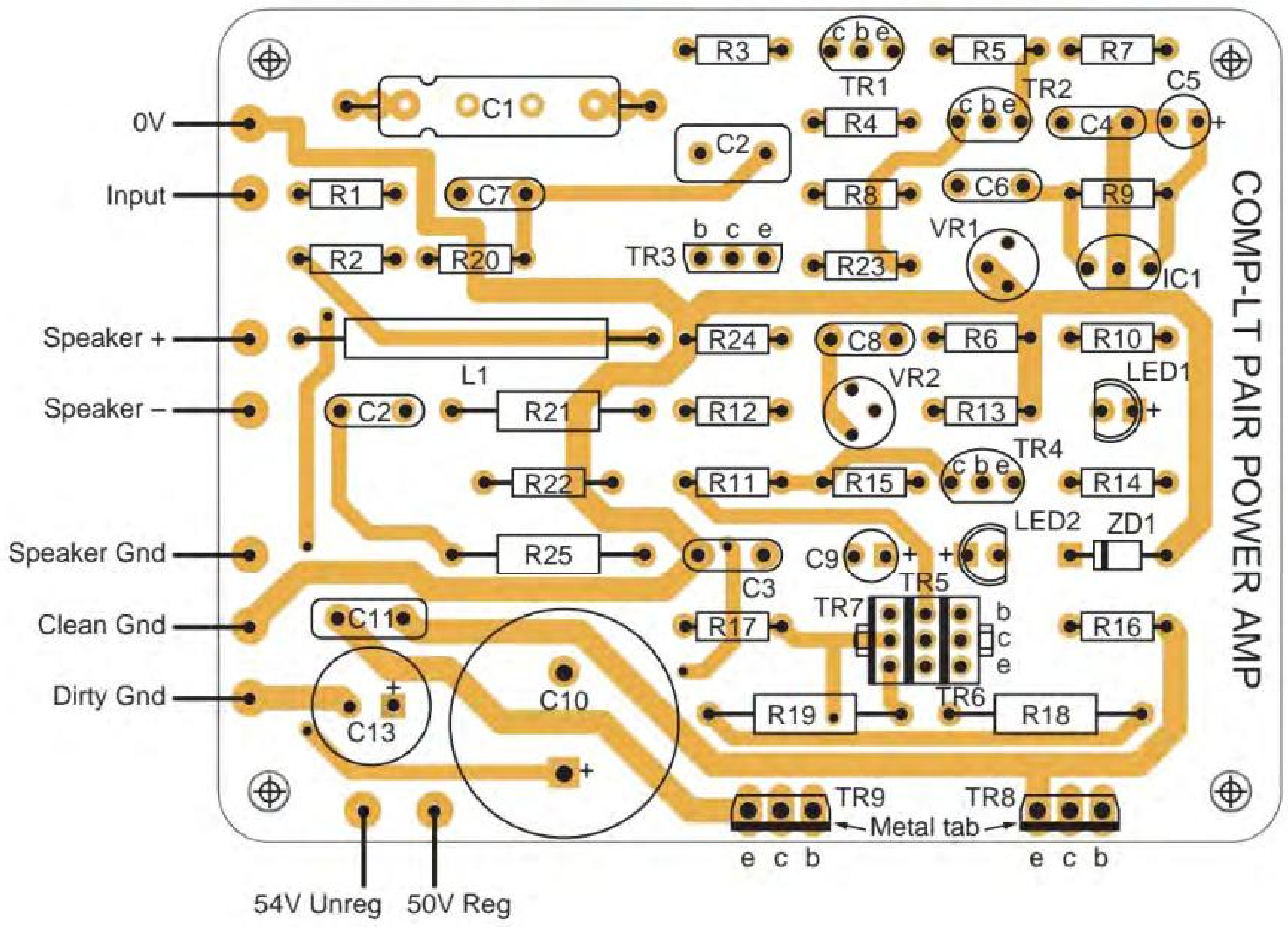

0V
Input
Speaker +
Speaker –
Speaker Gnd
Clean Gnd
Dirty Gnd
54V Unreg
50V Reg
COMP-LT PAIR POWER AMP
R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13 R14 R15 R16 R17 R18 R19 R20 R21 R22 R23 R24 R25
C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C13
TR1 TR2 TR3 TR4 TR5 TR6 TR7 TR8 TR9
VR1 VR2
IC1
LED1 LED2
ZD1
L1
Metal tab
e c b

Fig.4: the cascode amplifier PCB overlay.

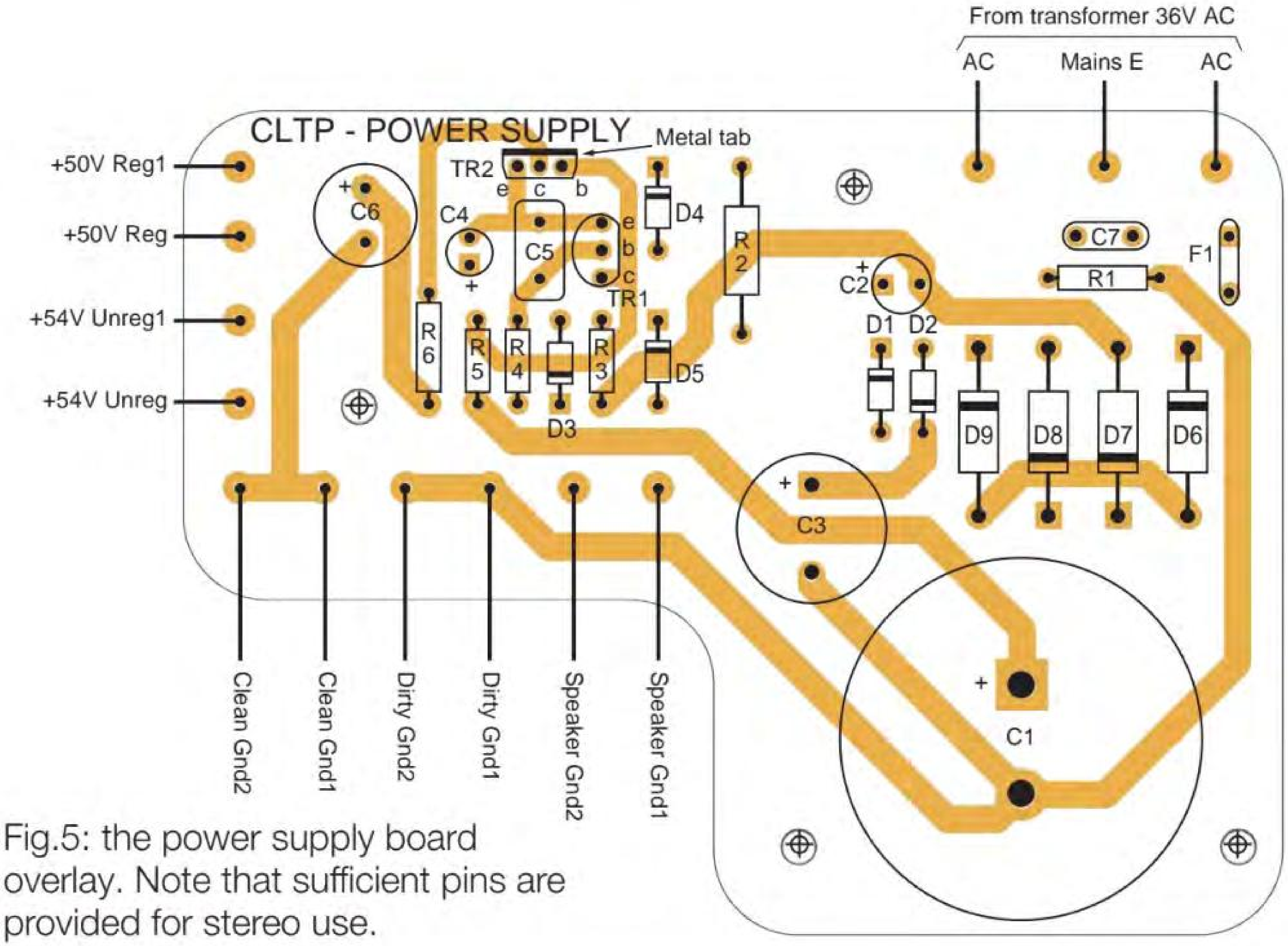

From transformer 36V AC
AC
Mains E
AC
CLTP - POWER SUPPLY
Metal tab
+50V Reg1
+50V Reg
+54V Unreg1
+54V Unreg
TR1 TR2
C1 C2 C3 C4 C5 C6 C7
R1 R2 R3 R4 R5 R6
D1 D2 D3 D4 D5 D6 D7 D8 D9
F1
Clean Gnd2
Clean Gnd1
Dirty Gnd2
Dirty Gnd1
Speaker Gnd2
Speaker Gnd1

Fig.5: the power supply board overlay. Note that sufficient pins are provided for stereo use.

version of the phase-splitter from Fig.2 is shown in Photo 2.

## Bridging amplifier impedances

The voltage loss on the discrete LTP phase splitter outputs could be reduced by making the power amplifiers non-inverting, giving a higher input impedance, thereby reducing the loading. This would be quite tricky with the PCB given here.

One of the great advantages of bridging is that the output voltage is double that of a single amplifier. For a given speaker impedance, say 8Ω, the output power is theoretically quadrupled. This is because each amplifier has to deliver twice the current as well as there being twice as many outputs.

The problem is that the speaker impedance 'seen' by the system is halved, ie, our 8Ω speaker looks like a 4Ω one (that's why the current doubles). This doubling of current can stress the amplifiers, increasing the distortion and dissipation. One way around this is to reduce the transformer voltage a bit and double the VAS stage (TR3) current.

In my case, this is great because I have loads of 15-0-15V 60VA toroidal transformers in stock. These should give around 50W RMS into 8Ω bridged for short periods with no stress. Another single amplifier could also be run for a tweeter amplifier, forming the basis of an excellent active speaker system.

When using two amplifiers in bridge mode, it is important to use a single joint power supply; strange distortion-inducing Earth loops can occur if separate power supplies are used with a common Earth.

An interesting effect of using Class-AB amplifiers in bridge mode is that the current pulses drawn from the supply are full-wave rectified versions of the signal rather than half-wave. In theory, this means that the smoothing capacitors have an easier time because there is an effective doubling of the frequency of the signal-derived ripple on the supply rail.

There is an advantage in dual-rail bridge systems in that no speaker current goes into the Earth; it's entirely confined to the power rails. In this case, where the amplifiers are single-rail, the 'negative' rail and speaker Earth are one and the same. I suspect there is little advantage in practice, because supply wiring impedances are going to be more significant.

**Ding dong**; the boards have arrived, so I can get the soldering iron out at last. Still, the LTP phase-splitter is a useful circuit block. Did you know it can form the basis of an all-pass filter or phase-shifter? Now there's an interesting idea.

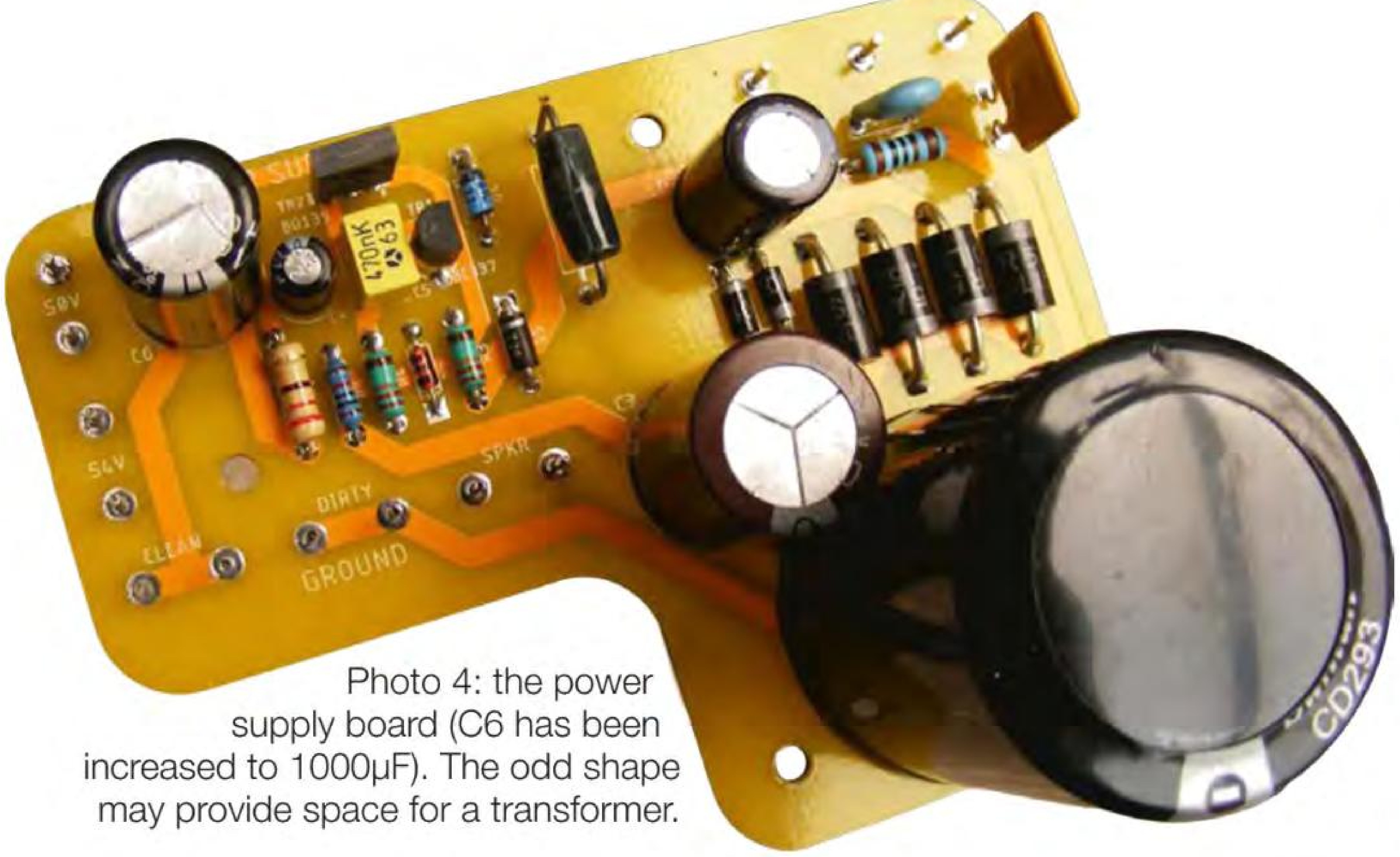


Photo 4: the power supply board (C6 has been increased to 1000µF). The odd shape may provide space for a transformer.

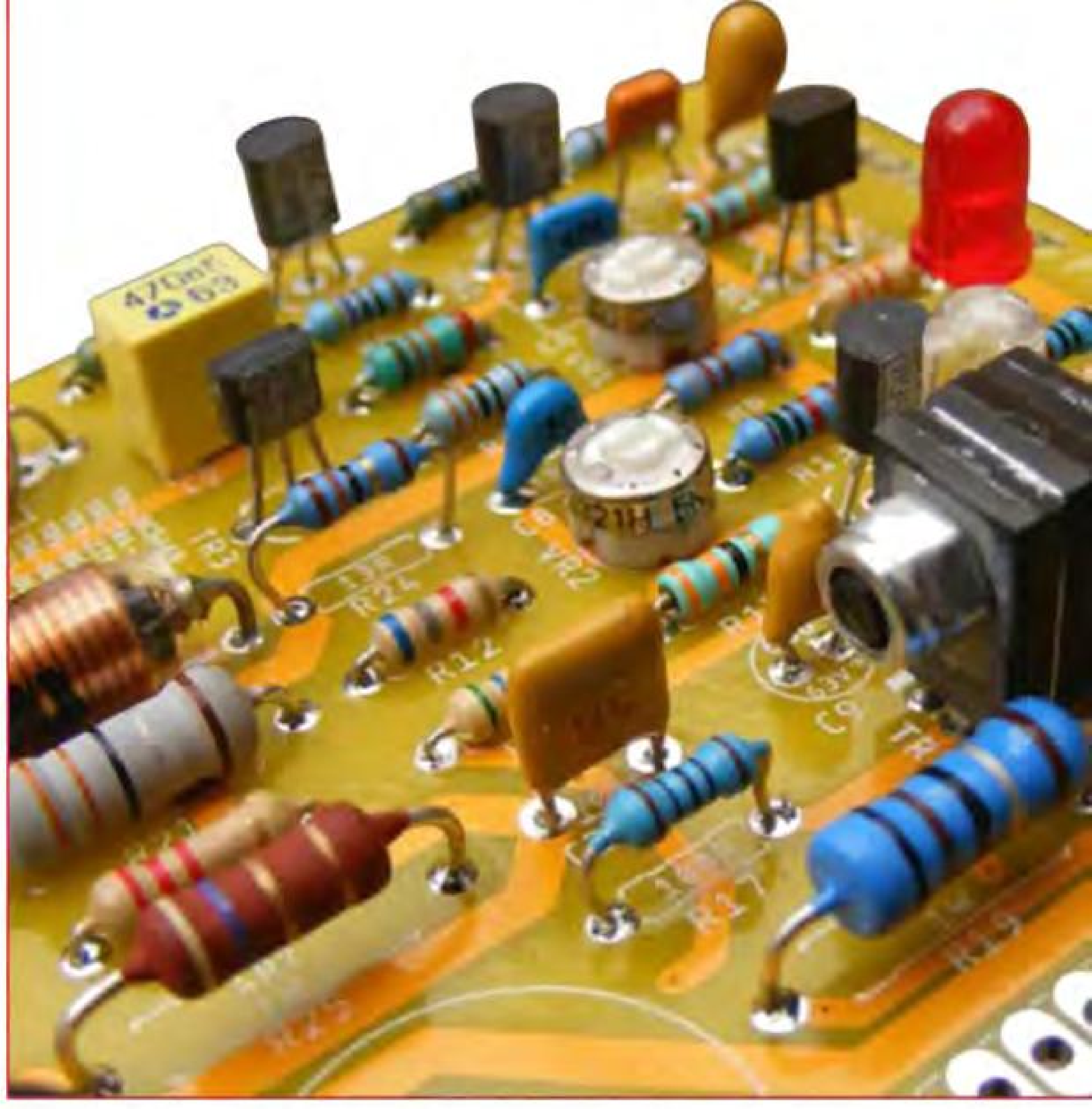


Photo 5: it's a good idea to space the resistors a few millimetres above the board, even if they normally run cool.

## Building the cascode power amp

The PCB overlays for the amplifier and power supply are shown in Figs.4 & 5.

When initiating component insertion, remember that putting electrolytic capacitors in the wrong way is more catastrophic in single-rail circuitry because of the full polarisation voltage. Reversal of the output electrolytic, C10, will cause a big explosion and take your loudspeaker and output transistors with it! Positive is denoted by the square pad on the PCB. The same goes for the LEDs.

The components are spaced out on the boards for educational use and experimentation, as shown in Photos 3 & 4.

It is a good idea to space some resistors a few millimetres above the board (Photo 5) in case they burn up during fault conditions, causing damage such as holes in the PCB and possible fires.

The Zobel resistor (R25), constant-current source resistor (R14), VAS emitter resistor (R24), driver transistor resistors (R16 & R17), output transistor resistors (R18 & R19) should be raised in all power amps for this reason.

It's quite reliable to do this on boards with plated-through holes (eg, double-sided PCBs). However on single-sided PCBs, it can result in broken pads and tracks if the components get pushed down, unless the leads are supported with special crimps or ceramic beads. The sealed Bourns-pattern TO5 trimmer resistors have very small holes; you can't ram normal skeleton presets in. These tend to crack up and oxidise, anyway.

## A transistor 'sandwich'

This is not lunch but a thermal construction where the driver transistors are bolted either side of the thermal sense ($V_{BE}$) transistor, TR5, as shown in Photo 6. Instead of butter, thermal paste is used to ensure good coupling. They are held together using a 12mm-long M3 machine screw with a lock nut. Note that the metal tabs all face the same way; left, towards the nut/output capacitor.

No insulating washers are needed because the inside of the device holes are insulated.

I had changed the original Zetex transistors used for the more common BD139/40 devices for the drivers, TR6 & TR7. TR5 can be any NPN TO-126 device. I used a BD135 because it was at hand.

I thought the original copper foil

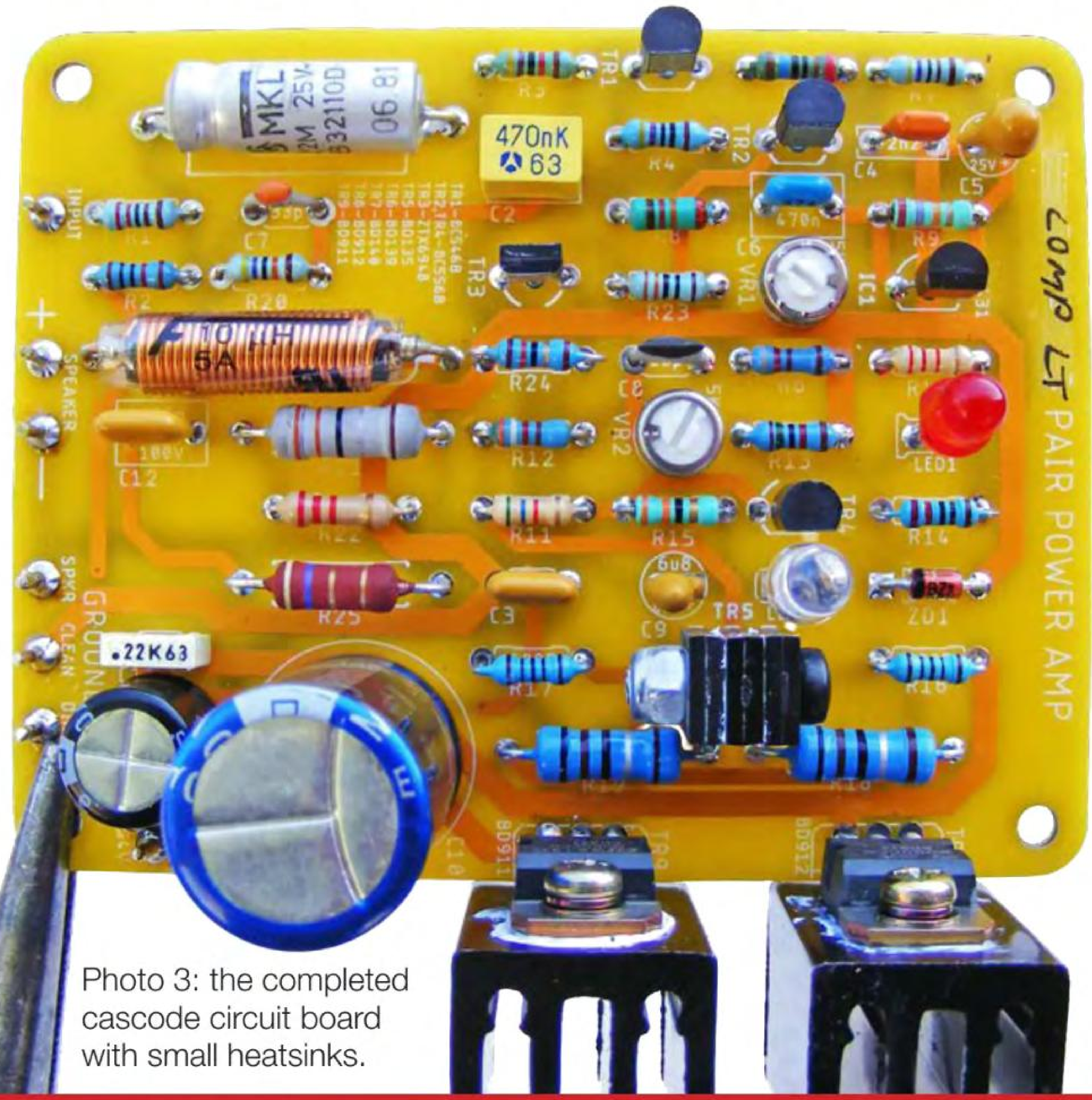


Photo 3: the completed cascode circuit board with small heatsinks.

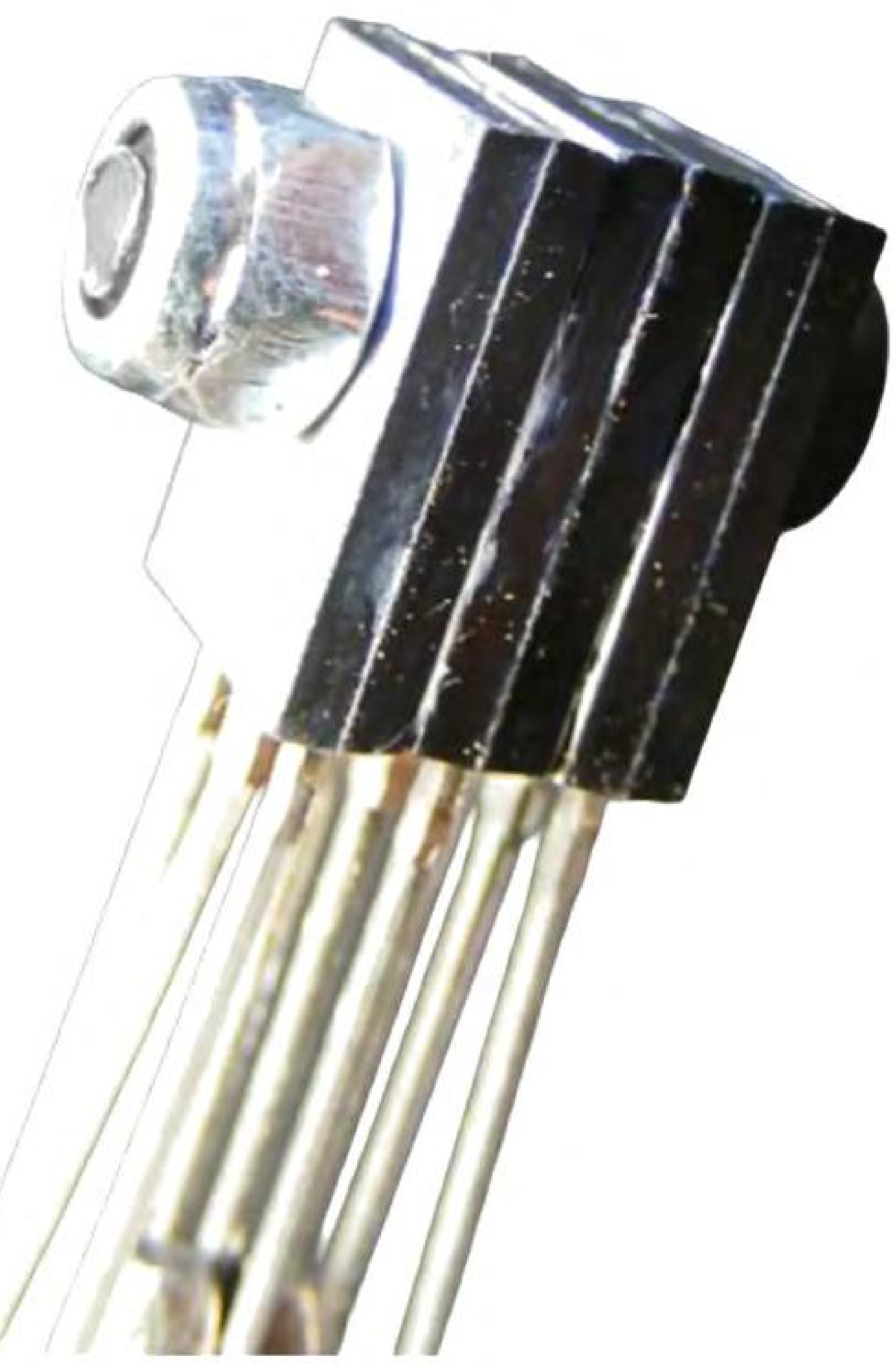


Photo 6: the 'transistor sandwich'. I thought it was the best thing in thermal compensation, but there were side effects.

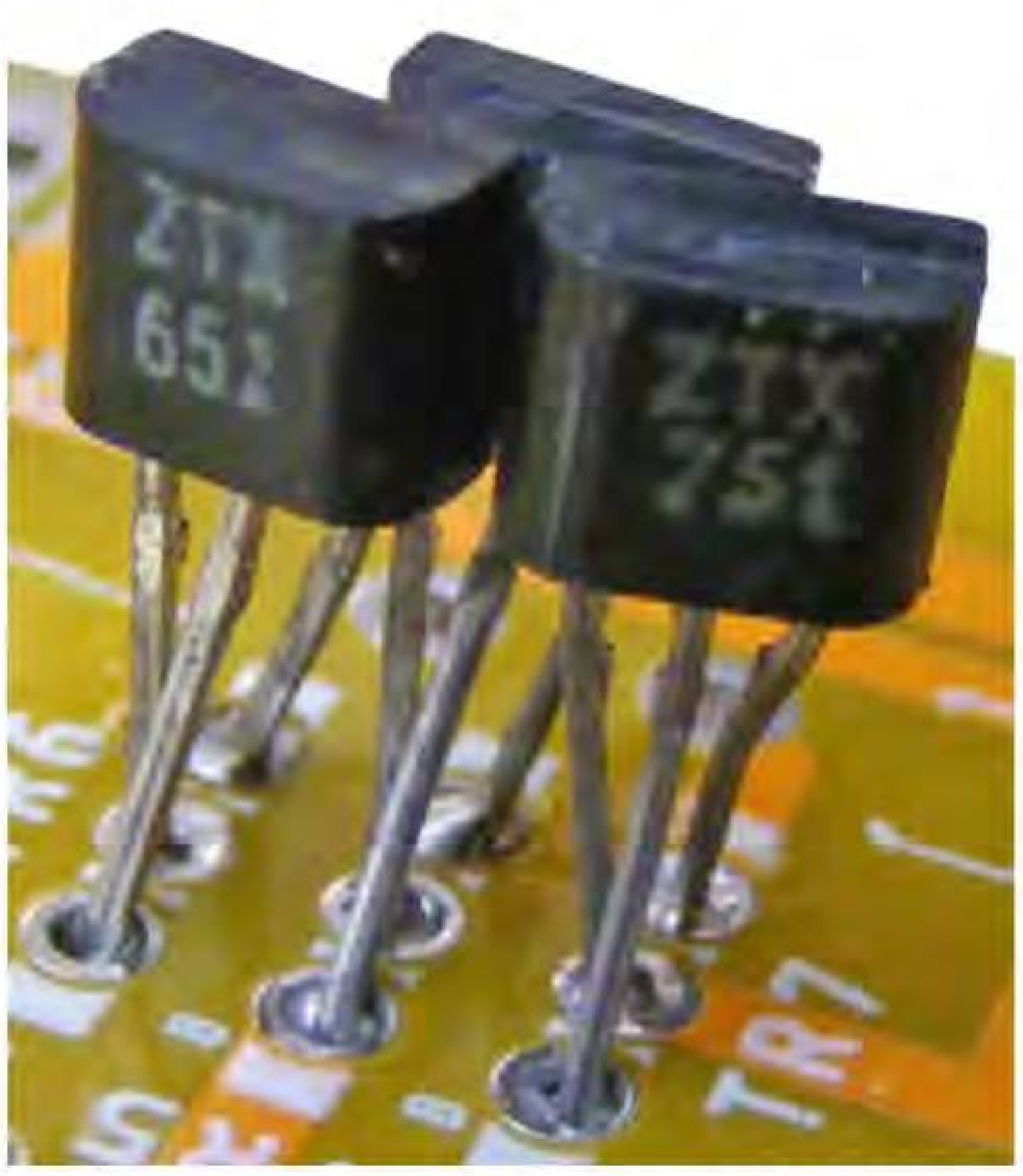


Photo 7: soldering the original ZTX transistors to the PCB. This isn't ideal, so I am designing a different board.

mounting arrangement was too tricky. If you want to use the original transistors, they can be fitted as shown in Photo 7, but because they are centre-base rather than centre-collector, a bit of 'leg crossing' is called for.

## Component substitutions

Transistor substitutions can be tricky with power amplifiers, and this proved to be no exception. Two component values had to be changed. Firstly, C8, the compensation capacitor, had to be increased to 56pF because the BD140 (TR7) is slower than the ZTX751, causing high-frequency oscillation.

Also, the BD135 has a slightly lower (0.69V) switch-on voltage than the ZTX300 (0.78V) used for TR5. This meant that R12 had to be changed to 3.9kΩ. Interestingly, the 'transistor sandwich' actually over-compensates, the thermal coupling being too good. This could be beneficial if monolithic Darlington transistors were used for the output stage.

Whenever substitutions on any power amp are made, it's essential to check for oscillations with a scope on the output.

Because there is less high-frequency open-loop gain due to the increased compensation, the HF distortion is worsened slightly from 0.02% to 0.03%, as shown in Fig.6. This could be fixed by using second-order compensation, but that's for another investigation. I didn't notice any subjective differences.

I've always agreed with Baxandall and Peter Walker (of Quad) that all competent amplifiers sound very similar up to overload.

## Simplifications

Output inductor L1 is only needed to prevent oscillation with capacitance from long twin-flex speaker leads. If the speaker leads are less than, say, 250mm (10 inches), such as in active speaker cabinets, it can be omitted and its position linked out on the PCB, along with resistor R21.

## Biasing

If the board is to be used with a regulated driver supply, the bias chain can be replaced by resistors and regulator IC1 omitted. In this case, the zener diode supply still has a bit of hum. Remember that the bias input is a signal pin as well, so any noise on here is amplified.

## Heatsinking

The output transistors are designed to be mounted onto a heatsink or a thick aluminium chassis. Insulating washers and bushes must be used since the metal tabs are at +23V. A thermal resistance no higher than about 3.4°C/W is needed.

## Power supply

The main reservoir capacitor C1 should be 63V, not 50V as originally specified. With some mains transformers, the rail voltage can rise to 55V off load. The capacitor is unlikely to short out because the electrode foil oxide is formed to the surge voltage rating (20% higher), but it's best to design conservatively.

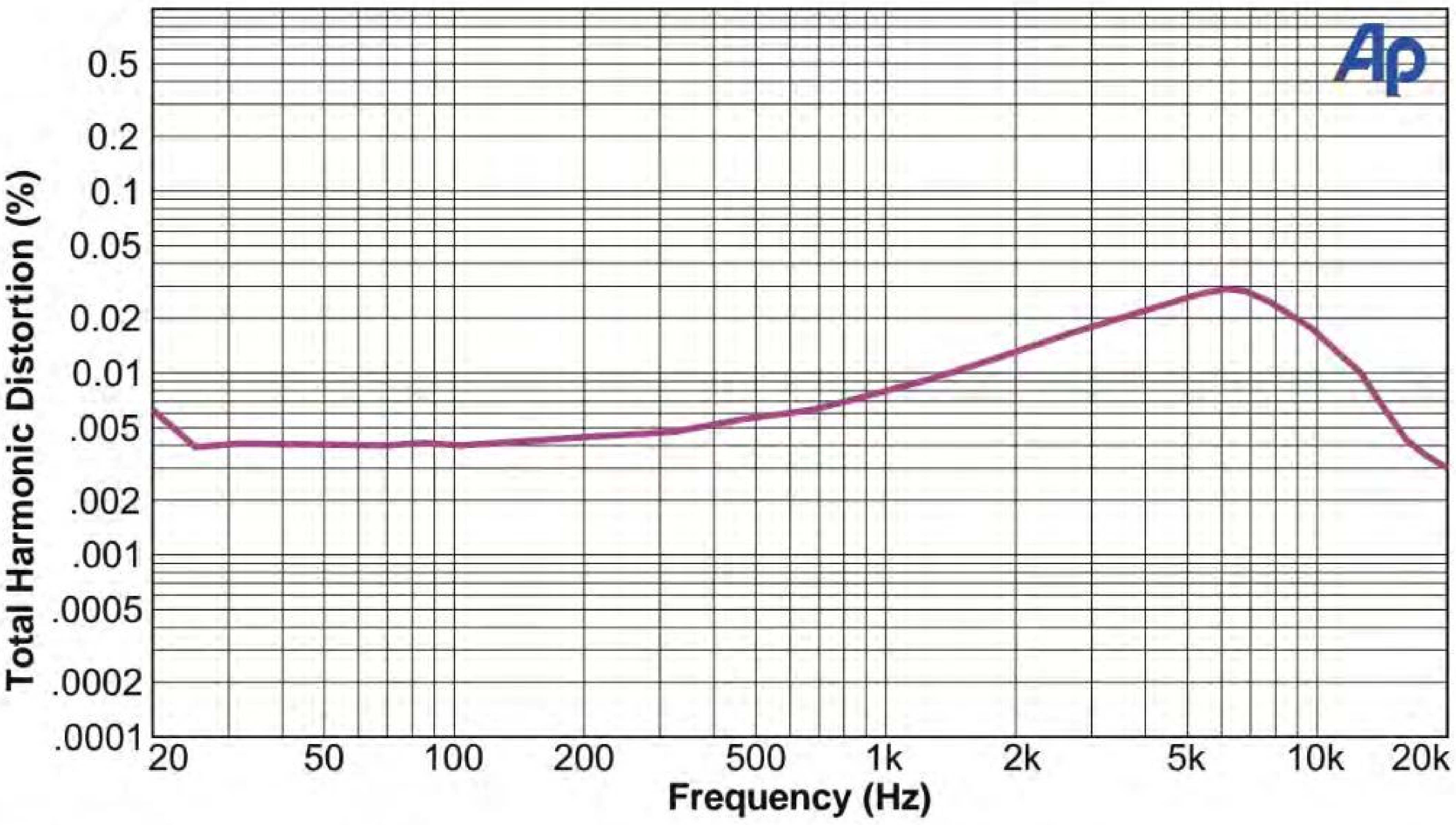


Fig.6: the distortion is slightly higher using BD139/40 driver transistors compared to the original ZTX751/ZTX651. The output is 12.7V peak-to-peak, 2.5W RMS into 8Ω.

## Wiring

The output transistors can be wired off-board with quite long leads, up to 200mm, before instability occurs. Note there are three Earth wires from the board. To keep impedances low, 16/0.2mm stranded wire is used for the high-current connections. The regulated supply and transistor base connections can be thinner, such as 7/0.1mm. The completed amplifier test jig is shown in Photo 8.

## Testing

The first thing to do is put the trimmers in the correct position. The bias, VR1, should be set mid-way and the quiescent current, VR2, fully anticlockwise to maximum resistance. No speaker or load is ever connected with initial power amp testing.

If you have a proper bench power supply that goes up to 50V (most only go up to 30V) with current limiting, one can take a cautious approach to testing with no blow-ups. If not, one can put 100Ω ¼W resistors in the power rails for protection. The smoke will soon tell you if there's a fault!

Remember that the output transistors can be un-powered by not connecting the 54V rail. This will enable low-level tests to be performed safely.

## DC tests

Once the DC conditions are fine, with the output at somewhere around half-rail, the bias can be set to 20V with VR1. This can be checked across C5; the output emitter resistors (R18 and R19) are a good test point for this. If you want to check the VAS current source, there should be just over 1V across R14.

The rail currents can next be checked with an ammeter connected in series with the power supply. The regulated 50V rail should draw around 7mA. The quiescent current can be set next, with an ammeter in series with the +54V rail. This should be zero until VR2 is advanced. There will be the usual point where it suddenly jumps up, so turn it slowly.

Set it to around 100mA, and as the 'transistor sandwich' warms up, it will drop to the required value of 15-30mA. It will take around 10 minutes to settle down. This could be sped up by adding heatsinking, but that's for the next board version. If you have a sensitive voltmeter, the voltage drop across R18 and R19 can be measured; it will only be a couple of millivolts.

## Signal testing

Now it's time to get the signal

generator out and give the amp a bit of a thrash. With no load, check it clips symmetrically and cleanly on a 1kHz sinewave. If it's okay, try it at say 20Hz and 10kHz. Next, try square waves, looking for excessive ringing.

### Load testing

The next test is to attach an 8Ω 30W load resistor. The output voltage will drop a couple of volts, but it should be possible to obtain over 40V peak-to-peak. VR1 can finally be trimmed to obtain symmetrical clipping. Only leave it powered for a minute or so at a time, or the resistor and heatsinks will get very hot.

I got a slight increase in power to 28W/42.5V peak-to-peak compared to the Veroboard version because I used schottky diodes in the bridge rectifier rather than bog-standard ones.

### Bridge testing

Now that we have two working boards, it will be interesting to explore bridge-mode operation and the distortion effects of things like wiring layout. Initial tests were not promising, with distortion at HF more than doubling, confirming my experience that bridge-mode amplifiers generally sound inferior.

It is possible to omit the output capacitors in a bridge amplifier, but do this at your peril, since both outputs are at +23V. A single short to ground and bang go the output transistors. I'm sure with further R&D, improvements will be made. Then we'll build them into a nice case, but I'm out of time for now. PE

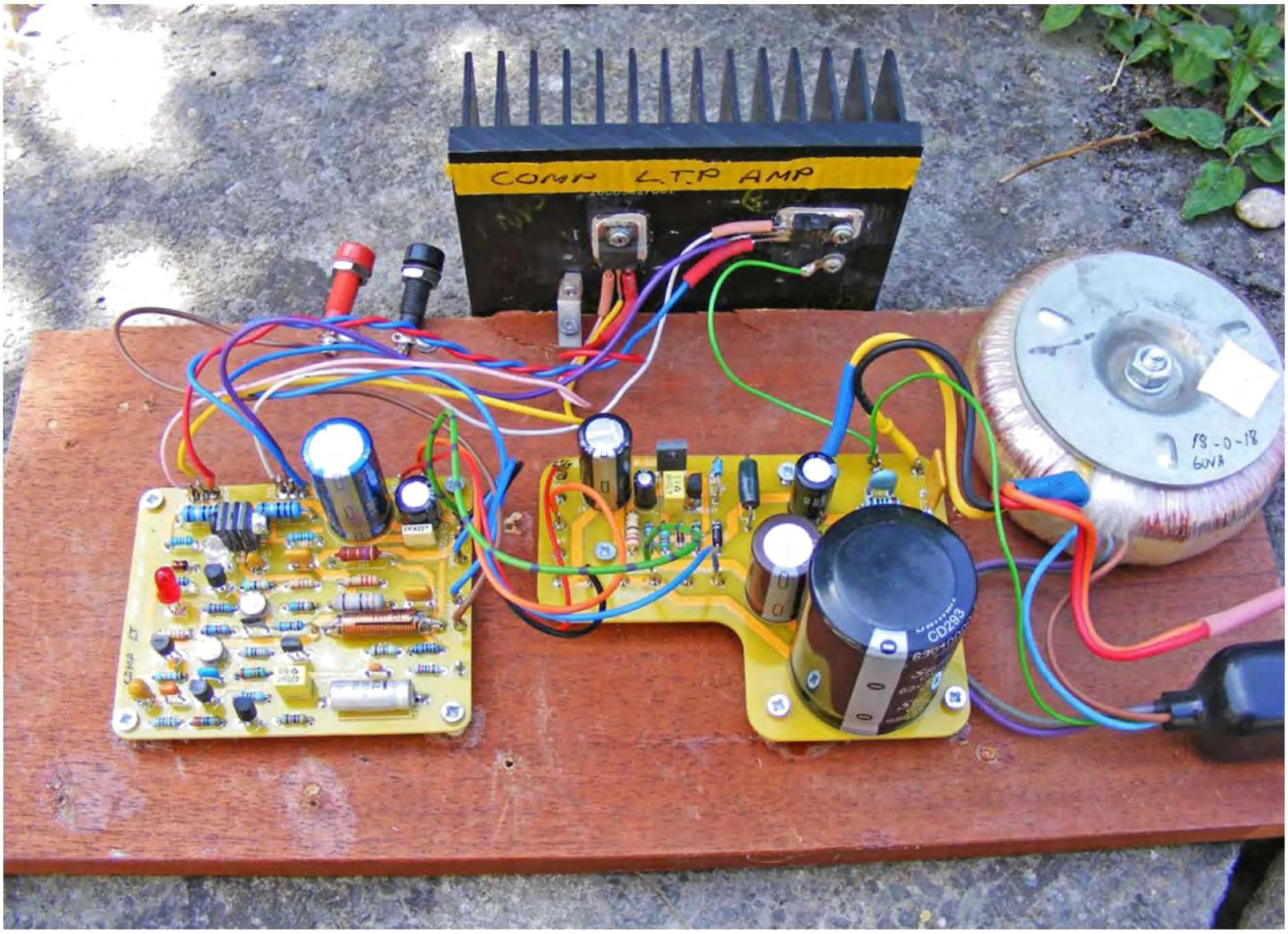


Photo 8: the test jig at the prototype PCB stage – less precarious than the old Veroboard version. A pretty enclosure awaits. "You always say that", says Kate, as she sweeps piles of mouldy circuits from the dining room table.

HAMMOND®

1591 ABS flame-retardant enclosures

Learn more:
hammondmfg.com/1591

uksales@hammfg.com • 01256 812812

# Practical Electronics PCB SERVICE

| PROJECT | CODE | PRICE |
|---|---|---|
| **SEPTEMBER 2025** | | |
| Automatic LQ Meter PCB | CSE240203A | £5.99 |
| Automatic LQ Meter front panel (black) | CSE240204A | £5.99 |
| Compact Hi-Fi Headphone Amplifier PCB (blue) | 01103241 | £4.99 |
| Discrete Ideal Bridge Rectifier TH PCB | 18108241 | £2.99 |
| Discrete Ideal Bridge Rectifier SMD PCB | 18108242 | £2.99 |
| Dual Rail Supply Protector PCB | 18109241 | £3.99 |
| Audio Out Cascode Amplifier PCB | CASCODEA | £4.99 |
| Audio Out Cascode Amplifier Power Supply PCB | CASCODEP | £4.99 |
| **AUGUST 2025** | | |
| Micromite Explore-40 PCB | 07106241 | £2.99 |
| Micromite Explore-40 Audio Breakout PCB | 07101222 | £2.99 |
| 8-Channel Learning IR Remote Receiver PCB | 15108241 | £5.99 |
| FlexiDice PCB set | 08107241-2 | £4.99 |
| **JULY 2025** | | |
| Compact OLED Clock & Timer PCB | 19101231 | £4.99 |
| 180-230V DC Motor Speed Controller PCB | 11104241 | £9.99 |
| Mains Power-Up Sequencer PCB (same as Feb '25) | 10108231 | £11.99 |
| **JUNE 2025** | | |
| Styloclone (case version) | 23106241 | £6.99 |
| Styloclone (standalone version) | 23106242 | £7.99 |
| Dual Mini LED Dice (TH LEDs, white solder mask) | 08103241W | £2.99 |
| Dual Mini LED Dice (TH LEDs, black solder mask) | 08103241B | £2.99 |
| Dual Mini LED Dice (SMD LEDs, white solder mask) | 08103242W | £2.99 |
| Dual Mini LED Dice (SMD LEDs, black solder mask) | 08103242B | £2.99 |
| Dual Hybrid Supply regulator board | 18107211 | £5.99 |
| Dual Hybrid Supply front panel control board | 18107212 | £2.99 |
| Dual Hybrid Supply CPU board | 01106193 | £3.99 |
| Dual Hybrid Supply LCD adaptor | 01106196 | £2.99 |
| **MAY 2025** | | |
| Skill Tester 9000 | 08101241 | £9.99 |
| USB-C Serial Adaptor (black solder mask) | 24106241 | £2.99 |
| DC Supply Protector (SMD adjustable version) | 08106241 | £3.99 |
| DC Supply Protector (TH adjustable version) | 08106242 | £3.99 |
| DC Supply Protector (TH fixed version) | 08106243 | £3.99 |
| **APRIL 2025** | | |
| ESR Test Tweezers PCB set | 04105241+ | £5.99 |
| Fan Speed Controller Mk2 | 10104241 | £4.99 |
| Calibrated Microphone (SMD version) | 01108231 | £3.99 |
| Calibrated Microphone (TH version) | 01108232 | £3.99 |
| **MARCH 2025** | | |
| Pico Gamer | 08104241 | £9.99 |
| Compact Frequency Divider | 04112231 | £4.99 |
| **FEBRUARY 2025** | | |
| Laser Communicator transmitter | 16102241 | £3.99 |
| Laser Communicator receiver | 16102242 | £3.99 |
| Model Railway Points Controller transmitter | 09101241 | £3.99 |
| Model Railway Points Controller receiver | 09101242 | £2.99 |
| Mains Power-Up Sequencer | 10108231 | £11.99 |
| **JANUARY 2025** | | |
| USB to PS/2 Converter for VGA PicoMite | 07111231 | £3.99 |
| ps2x2pico USB to PS/2 Converter | 07111232 | £3.99 |
| Compact Microphone Preamplifier (case version) | 01110231 | £5.99 |
| Compact Microphone Preamplifier (integrated version) | 01110232 | £5.99 |

| PROJECT | CODE | PRICE |
|---|---|---|
| **DECEMBER 2024** | | |
| Raspberry Pi Clock Radio PCB pair | 19101241/2 | £12.99 |
| VGA PicoMite (blue) | 07107221 | £5.99 |
| Secure Remote Switch receiver | 10109231 | £4.99 |
| Secure Remote Switch transmitter (module version) | 10109232 | £2.99 |
| Secure Remote Switch transmitter (discrete version) | 10109233 | £2.99 |
| **NOVEMBER 2024** | | |
| Ideal Diode Bridge Rectifiers (28mm square) | 18101241 | £3.99 |
| Ideal Diode Bridge Rectifiers (22mm square) | 18101242 | £3.99 |
| Ideal Diode Bridge Rectifiers (4-pin SIL) | 18101243 | £3.99 |
| Ideal Diode Bridge Rectifiers (tiny SOT-23) | 18101244 | £3.99 |
| Ideal Diode Bridge Rectifiers (standalone SMD) | 18101245 | £4.99 |
| Ideal Diode Bridge Rectifiers (standalone TH; 2oz copper) | 18101246 | £6.99 |
| Multi-channel Volume Control (volume PCB) | 01111221 | £4.99 |
| Multi-channel Volume Control (control PCB) | 01111222 | £4.99 |
| Multi-channel Volume Control (OLED PCB, black) | 01111223 | £4.99 |
| **OCTOBER 2024** | | |
| TQFP-32 Programming Adaptor | 24108231 | £4.99 |
| TQFP-44 Programming Adaptor | 24108232 | £4.99 |
| TQFP-48 Programming Adaptor | 24108233 | £4.99 |
| TQFP-64 Programming Adaptor | 24108234 | £4.99 |
| 2m VHF FM Test Signal Generator | 06107231 | £4.99 |
| **SEPTEMBER 2024** | | |
| PIC Programming Adaptor | 24105231 | £4.99 |
| Salad Bowl Speaker crossover | 01109231 | £7.99 |
| **AUGUST 2024** | | |
| Arduino ESR Meter combined L/C-ESR | 04106182 | £9.99 |
| Arduino ESR Meter separate ESR | 04106183 | £7.99 |
| Watering System Controller | 15110231 | £11.99 |
| **JULY 2024** | | |
| Dynamic NFC Tag (22 × 31mm) | 06101231 | £5.95 |
| Dynamic NFC Tag (86 × 54mm) | 06101232 | £7.95 |
| Reciprocal Frequency Counter | CSE230101C | £8.95 |
| Reciprocal Frequency Counter front panel (black mask) | CSE230102 | £7.95 |
| Pi Pico-based Thermal Camera | 04105231 | £6.95 |
| **JUNE 2024** | | |
| Loudspeaker Test Jig | 04106231 | £12.95 |
| **MAY 2024** | | |
| GPS-Disciplined Oscillator | 04103231 | £12.95 |
| Dual RF Amplifier | CSE220602A | £5.95 |
| Songbird | 08103231 | £7.95 |
| **APRIL 2024** | | |
| Basic RF Signal Generator | CSE221001 | £8.99 |
| Basic RF Signal Generator front panel (black solder mask) | CSE220902B | £9.99 |
| Silicon Chirp Cricket | 08101231 | £7.99 |
| Test Bench 'Swiss Army Knife' | 04110221 | £9.99 |
| **MARCH 2024** | | |
| Digital Volume Control Potentiometer SMD version | 01101231 | £8.95 |
| Digital Volume Control Potentiometer through-hole version | 01101232 | £9.95 |
| Advanced SMD Test Tweezers body | 04106221 | £6.95 |
| Advanced SMD Test Tweezers legs (pair) | 04106212 | £7.95 |
| **FEBRUARY 2024** | | |
| Active Mains Soft Starter | 10110221 | £10.95 |

PCBs for most recent *PE/EPE* constructional projects are available. From the July 2013 issue onwards, PCBs with eight-digit codes have silk screen overlays and, where applicable, are double-sided, have plated-through holes, and solder mask. They are similar to photos in the project articles. Earlier PCBs are likely to be more basic and may not include silk screen overlay, be single-sided, lack plated-through holes and solder mask.

Always check price and availability in the latest issue or online. A large number of older boards are listed for ordering on our website.

**In most cases we do not supply kits or components for our projects. For older projects it is important to check the availability of all components before purchasing PCBs.**

Magazine back issues are available – see https://www.electronpublishing.com/product-category/pe-back-issues-print-and-digital/

# Double-sided | plated-through holes | solder mask

| PROJECT | CODE | PRICE |
|---|---|---|
| **JANUARY 2024** | | |
| Q Meter | CSE220701 | £8.95 |
| Q Meter front panel (black solder mask) | CSE220704 | £7.95 |
| Raspberry Pi Pico W BackPack | 07101221 | £8.95 |
| **DECEMBER 2023** | | |
| Digital Boost Regulator | 24110224 | £8.95 |
| Dual-Channel Power Supply for Breadboards | 04112221 | £11.95 |
| Display Adaptor for the Breadboard PSU | 04112222 | £9.95 |
| **NOVEMBER 2023** | | |
| LC Meter Mk3 | CSE220503C | £9.95 |
| DC Supply Filter for vehicles | 08108221 | £8.95 |
| Discrete Audio Op Amp PCB | AO1-JUL23 | £9.95 |
| Discrete Audio Op Amp PCB + essential components | AO2-JUL23 | £17.95 |
| **OCTOBER 2023** | | |
| Buck/Boost Charger Adaptor | 14108221 | £9.95 |
| PIC Breakout Board for SOIC parts | 24110225 | £5.95 |
| PIC Breakout Board for DIP parts | 24110222 | £5.95 |
| AVR64DD32 Breakout board | 24110223 | £5.95 |
| Automatic Train Controller | 09109221 | £6.95 |
| Chuff Sound module | 09109222 | £6.95 |
| **SEPTEMBER 2023** | | |
| Mini LED Driver | 16106221 | £7.95 |
| New GPS-Synchronised Clock | 19109221 | £9.95 |
| **AUGUST 2023** | | |
| 110dB RF Attenuator | CSE211003 | £9.95 |
| Universal Battery Charge Controller (2023 update) | 14107192 | £9.95 |
| Wide-Range OhmMeter | 04109221 | £12.95 |
| **JULY 2023** | | |
| Multimeter Checker/Calibrator | 04107221 | £10.95 |
| **JUNE 2023** | | |
| Arduino Programmable Load | 04105221 | £8.95 |
| Buck-Boost LED Driver | 16103221 | £8.95 |
| **MAY 2023** | | |
| Precision AM-FM DDS Signal Generator | CSE211002 | £14.95 |
| Improved SMD Test Tweezers programmed PIC | 0410621PIC | £12.95 |
| AO 2x-dual-to-quad through-hole (pack of five) | AO1-MAY23 | £6.95 |
| AO 2x-dual-to-quad SMD (pack of five) | AO2-MAY23 | £6.95 |
| AO 2x-single-to-dual through-hole (pack of five) | AO3-MAY23 | £6.95 |
| AO 2x-single-to-dual SMD (pack of five) | AO4-MAY23 | £6.95 |
| **APRIL 2023** | | |
| 500W Amplifier Module | see p.22, April 2023 | |
| Clipping Indicator (per channel) | 01112211 | £7.95 |
| CD Welder Power Supply (one needed) | 29103221 | £9.95 |
| CD Welder Controller (one needed) | 29103222 | £9.95 |
| CD Welder Energy Storage module (several needed) | 29103223 | £7.95 |
| AO Universal Dual Op Amp Board | AO1-APR23 | £9.95 |
| AO Stereo RIAA precision passives kit for dual op amp | AO2-APR23 | £8.95 |
| **MARCH 2023** | | |
| Pico BackPack | 07101221 | £9.95 |
| Semaphore Signal (controller) | 09103221 | £7.95 |
| Semaphore Signal (blade) | 09103222 | £5.95 |
| **FEBRUARY 2023** | | |
| Tesla Coil driver board | 26102221 | £9.95 |
| Tesla Coil potentiometer board | 26102222 | £5.95 |
| Cooling Fan Controller & Loudspeaker Protector | 01102221 | £8.95 |
| Remote Gate Controller | 11009121 | £12.95 |

| PROJECT | CODE | PRICE |
|---|---|---|
| **JANUARY 2023** | | |
| Classic LED Metronome – 8-LED | 23111211 | £7.95 |
| Classic LED Metronome – 10-LED | 23111212 | £8.95 |
| Multi-Channel Speaker Protector – 6 channel | 01101221 | £9.95 |
| Multi-Channel Speaker Protector – 4 channel | 01101222 | £7.95 |
| Remote Control Range Extender – IR-to-UHF | 15109212 | £5.95 |
| Remote Control Range Extender – UHF-to-IR | 15109211 | £7.95 |
| AO Universal Single Op Amp Board | AO1-JAN23 | £7.95 |
| **DECEMBER 2022** | | |
| Hummingbird Amplifier | 01111211 | £9.95 |
| SMD Trainer PCB | 29106211A | £8.95 |
| **NOVEMBER 2022** | | |
| USB Cable tester – main PCB | 04108211 | £12.95 |
| USB Cable tester – front panel | 04108212 | £5.95 |
| USB Cable tester – optional panel | SC5970 | £5.95 |
| Model Railway Carriage Lights – PCB | 09109211 | £6.95 |
| AO trnsfmr PCB – standard VTX-A range | VTX-101-007 | £6.95 |
| AO trnsfmr PCB – dual-outline VTX102-3001/101-3002 | VTX-Dual | £6.95 |
| **OCTOBER 2022** | | |
| SMD Test Tweezers – PCB and pair of tweezer arms | 04106211-2 | £11.95 |
| SMD Test Tweezers – programmed PIC12F1572-I/SN | 0410621A | £7.95 |
| Tele-com | 12110211 | £12.95 |

**For the many earlier PCBs that we stock please see the PE website: www.electronpublishing.com**

## PE/EPE PCB SERVICE

| Order Code | Project | Quantity | Price |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Name ..............................................

Address ..............................................

..............................................

Tel ..............................................

Email ..............................................

Card No ..............................................

Valid From .................. Expiry Date ................

Card Security No ..........

**You can also order PCBs by phone, email or via the shop on our website: www.electronpublishing.com**

**No need to cut your issue – a copy of this form is just as good!**

All prices include VAT and UK p&p. Add £4 per *project* for post to Europe; £5 per *project* outside Europe.

Orders and payment should be sent to:

**Practical Electronics, Electron Publishing Ltd**
**113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU**
**Tel 07918 614662 Email: pe@pemag.au**
**Online Shop: www.electronpublishing.com**

Sorry, cheques cannot be accepted as we are based in Australia and our UK bank account does not allow cheque deposits.

**NOTE:** Most boards are in stock and sent within seven days of order receipt. Please allow up to 28 days delivery if we need to restock.

# Practical Electronics PCB SERVICE

| PROJECT | CODE | PRICE |
|---|---|---|
| **SEPTEMBER 2022** | | |
| Touchscreen Digital Preamp – main board | 01103191 | £12.95 |
| Touchscreen Digital Preamp – adaptor board pair | 01103192 | £5.95 |
| 20A DC Motor Speed Controller | 11006211 | £9.95 |
| **AUGUST 2022** | | |
| Multi-purpose Battery Manager – I/O Expander | 11104212 | £5.95 |
| Multi-purpose Battery Manager – Switch Module | 11104211 | £8.95 |
| Simple MIDI Music Keyboard (for 8 switches) | 23101213 | £6.95 |
| Nano Pong | 08105212 | £7.95 |
| **JULY 2022** | | |
| Silicon Labs AM/FM/SW Radio | CSE210301C | £10.95 |
| Level Crossing Controller | 09108211 | £6.95 |
| **JUNE 2022** | | |
| Full-wave Motor Speed Controller | 1010221 | £8.95 |
| PIC Programming Helper for 8-pin PICs only | 24106211 | £7.95 |
| PIC Programming Helper for 8, 14 or 20-pin PICs | 24106212 | £10.95 |
| Advanced GPS Computer | 05102211 | £9.95 |
| **MAY 2022** | | |
| Bus board PCB for Analogue Vocoder | AO1-MAY22 | £10.95 |
| Complete set of 14 PCBs for Analogue Vocoder | AO2-MAY22 | £97.95 |
| Programmed EEPROM for Digital FX Unit | FX1-MAY22 | £10.95 |
| Programmed PIC for Digital FX Unit w/ potentiometer | FX2-MAY22 | £8.95 |
| **APRIL 2022** | | |
| 64-key MIDI Matrix shield | 23101211 | £8.95 |
| 64-key MIDI Matrix switch board | 23101212 | £11.95 |
| High-current Battery Balancer | 14102211 | £10.95 |
| Digital FX Unit – using potentiometer | 01102211 | £9.95 |
| Digital FX Unit – using BCD switch | 01102212 | £9.95 |
| Universal Audio PSU | AO1-APR22 | £11.95 |
| **MARCH 2022** | | |
| Mini Isolated Serial Link | 24102211 | £5.95 |
| Busy Loo Indicator | 16112201 | £5.95 |
| Analogue Vocoder – Band-pass filter board | AO1-MAR22 | £9.95 |
| Analogue Vocoder – HP/LP filter board | AO2-MAR22 | £9.95 |
| **FEBRUARY 2022** | | |
| Arduino-based Power Supply | 18106201 | £9.95 |
| Battery Monitor Logger | 11106201 | £10.95 |
| Electronic Wind Chimes | 23011201 | £10.95 |
| Analogue Vocoder – Driver Amplifier | AO-FEB22 | £8.95 |
| **JANUARY 2022** | | |
| Vintage battery Radio Li-ion Power Supply | 11111201 | £9.95 |
| MiniHeart: A Miniature Heartbeat Simulator | 01109201 | £8.95 |
| **DECEMBER 2021** | | |
| AM/FM/SW Digital Receiver | CSE200902A | £13.95 |
| Balanced Input and Attenuator for USB CODEC | 01106202 | £11.95 |
| **NOVEMBER 2021** | | |
| Dual Battery Lifesaver | 11111202 | £6.95 |
| **OCTOBER 2021** | | |
| Mini Wi-Fi LCD BackPack | 24106201 | £8.95 |

| PROJECT | CODE | PRICE |
|---|---|---|
| **SEPTEMBER 2021** | | |
| USB SuperCodec PCB | 01106201 | £14.95 |
| Audio DDS Oscillator PCB | 01110201 | £5.95 |
| Audio DDS Oscillator rotary encoder | 01110201-ENC | 6.95 |
| Programming Adaptor Board for Audio DDS Oscillator | 01110202 | £5.95 |
| High-power Ultrasonic Cleaner main PCB | 04105201 | £14.95 |
| High-power Ultrasonic Cleaner front-panel PCB | 04105202 | |
| Night Keeper Lighthouse PCB | 08110201 | £5.95 |
| **AUGUST 2021** | | |
| Ol' Timer PCB | 19104201 | £11.95 |
| Ol' Timer 8x8 RGB LED module using WS2812B | 19104201-88 | £8.95 |
| Ol' Timer set of acrylic case pieces and spacer | 19104201-ACR | £8.75 |
| Ol' Timer DS3231 RTC IC wide SOIC-16 | 19104201-RTC | £5.95 |
| Wideband Digital RF Power Meter | 04106201 | £9.75 |
| Switchmode 78xx regulators ***(PACK of 5!)*** | 18105201 | £7.95 |
| Cool Beans SMAD display | CB-AUG21 | £11.95 |
| **JULY 2021** | | |
| ATtiny816 Breakout / Dev Board w/ Capacitive Touch | 24110181 | £9.75 |
| IR Remote Control Assistant (Jaycar version) | 15005201 | £8.95 |
| IR Remote Control Assistant (Altronics version) | 15005202 | £8.95 |
| PIC18F Development Board | PNM-JUL21 | £12.95 |
| Microphone Preamplifier | AO-JUL21 | £11.95 |
| **JUNE 2021** | | |
| Roadies' Test Signal Generator (surface-mount ver.) | 01005201 | £8.95 |
| Roadies' Test Signal Generator (through-hole version) | 01005202 | £9.95 |
| Touchscreen Wide-range RCL Box (Resistor module) | 04104201 | £18.95 |
| Touchscreen Wide-range RCL Box (Ind/Cap module) | 04104202 | |
| KickStart Part 3 – Gyrator-based Audio Filter | KS3-2021 | £7.95 |
| **MAY 2021** | | |
| 7-Band Equaliser (Mono) | 01104201 | £8.95 |
| 7-Band Equaliser (Stereo) | 01104202 | £10.95 |
| Car Altimeter | 05105201 | £7.95 |
| **APRIL 2021** | | |
| Reflow Oven – DSP Active Crossover (CPU) | 01106193 | £19.95 |
| Reflow Oven – DSP Active Crossover (Front panel) | 01106195 | |
| Reflow Oven – DSP Active Crossover (LCD) | 01106196 | |
| Frequency Reference Signal Distributor | CSE200103 | £8.95 |
| **MARCH 2021** | | |
| Nutube Guitar Effects Pedal | 01102201 | £12.95 |
| Programmable Thermal Regulator (Peltier Interface) | 21109181 | £18.95 |
| Programmable Thermal Regulator (Peltier Driver) | 21109182 | |
| Tunable HF Preamp | CSE190502 | £8.95 |
| **FEBRUARY 2021** | | |
| 4G Remote Monitoring | 27111191 | £9.95 |
| **JANUARY 2021** | | |
| Nutube Valve Preamplifier | 01112191 | £12.95 |
| Arduino DCC Controller | 09207181 | £10.95 |
| **DECEMBER 2020** | | |
| Pseudo-Random Sequence Generator | 16106191 | £7.95 |
| Clever Charger | 14107191 | £11.95 |
| PE Theremin Amplifier | AO-1220-01 | £8.95 |
| **NOVEMBER 2020** | | |
| LED Christmas Tree (1 off) | 16107181-1 | £6.95 |
| USB/SPI Interface Board | 16107182 | £5.95 |
| 45V/8A Power Supply PCB *plus* acrylic spacer | 18111181 | £14.95 |
| 45V/8A Power Supply five-way display bezel | 18111181-BZ | £3.95 |
| Five-way LCD Panel Meter/Display | 18111182 | £7.95 |

Silicon Chip kits & parts for projects in this issue (posted from Australia):

| | | |
|---|---|---|
| Automatic LQ Meter kit | https://siliconchip.au/Shop/20/6939 | AUD $120 (about £60.00) + P&P |
| Compact Hi-Fi Headphone Amplifier kit (including case) | https://siliconchip.au/Shop/20/6885 | AUD $84 (about £42.00) + P&P |
| Discrete Ideal Bridge Rectifier TH kit | https://siliconchip.au/Shop/20/6987 | AUD $36 (about £18.00) + P&P |
| Discrete Ideal Bridge Rectifier SMD kit | https://siliconchip.au/Shop/20/6988 | AUD $33 (about £16.50) + P&P |
| Dual Rail Load Protector partial kit | https://siliconchip.au/Shop/20/7366 | AUD $60 (about £30.00) + P&P |

20% VAT is included in the above prices (it will be removed for customers outside the UK). Postage & packing cost is typically around £12-15. Please allow 1-2 weeks for delivery (longer in some cases).

# CLASSIFIED ADVERTISING

## Practical Electronics

If you want your advertisements to be seen by the largest readership at the most economical price then our classified page offers excellent value.

The rate for semi-display space is £10 per centimetre high, with a minimum height of 2·5cm. All semi-display advertisements have a width of 5.5cm. The prepaid rate for classified adverts is 40p per word (minimum 12 words).

For rates, further information on display and classified advertising or to book an ad, please email us at **pe@pemag.au**

**Practical Electronics reaches more UK readers than any other UK monthly hobby electronics magazine. Our sales figures prove it.**

**We have been the leading monthly magazine in this market for more than thirty years.**

**W5SWL Electronics**

Technical Products and Small Component Parts, RF and Audio Connectors, Hobby Electronics, Ham Radio to Microwave Products, General Hobby Supplies, Technical Books, Tools, and More. **Located in the United States.**

Visit **www.W5SWL.com**

**COMPONENTS AS JOB LOT**
Large stock. Transistors, Diodes, Logic, Memory and Analogue parts. 1000s of passive items. Catalogued and sorted. 'New Old Stock' in racks. Email: rojobarn@gmail.com

**BOWOOD ELECTRONICS LTD**
*Suppliers of Electronic Components*
**www.bowood-electronics.co.uk**
Unit 10, Boythorpe Business Park, Dock Walk, Chesterfield, Derbyshire S40 2QR. Sales: 01246 200 222
Send large letter stamp for Catalogue

**MISCELLANEOUS**

**VALVES AND ALLIED COMPONENTS?**
For free stock list and/or advice, please contact me: geoffdavies337@gmail.com Telephone: 01788 574774

**PIC DEVELOPMENT KITS, DTMF kits** and modules, CTCSS Encoder and Decoder/Display kits.
Visit www.cstech.co.uk

**COAST ELECTRONICS**
BREAKOUTS-COMPONENTS-CONTRACT DESIGN-3D PRINTER PARTS-MUSICAL-MICROCONTROLLERS
**WWW.COASTELECTRONICS.CO.UK**

**Electrical Industries Charity (EIC)**
We help people working in the electrical, electronics and energy community as well as their family members and retirees.
We use workplace programmes that give the industry access to financial grants and a comprehensive range of free and confidential services.
**www.electricalcharity.org**
electrical industries charity

# ADVERTISING INDEX

AOSHOP .......................................... 55
BACK ISSUE PDFs.................................. 68, 69
BASIC MICRO .......................................... 17
ELECTRONICS TEACH-IN BUNDLES ...................... 15
ENIGMA COMPONENTS SHOP ............................ 25
HAMMOND MANUFACTURING............................. 75
MICROCHIP TECHNOLOGY ................ Outer back cover
PCBWAY.................................. Inner front cover
PO LABS .............................................. 54
TAG-CONNECT ......................................... 25

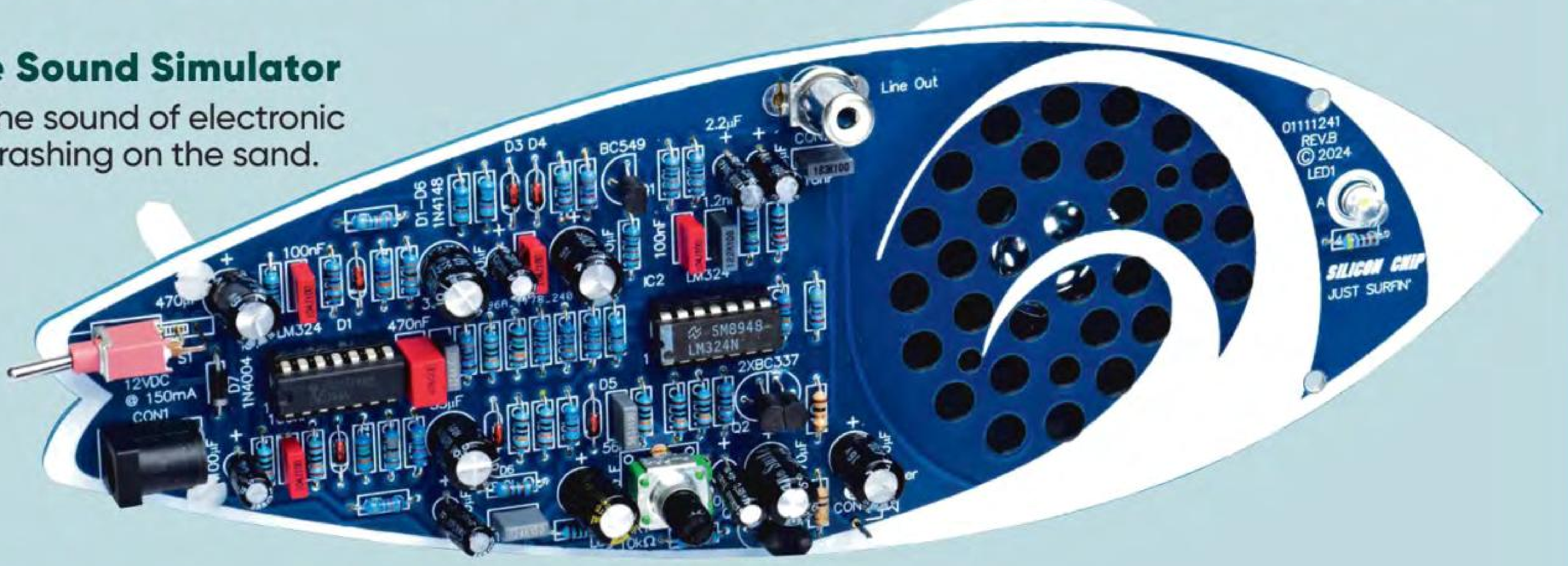
**Advertising mailing address**
Electron Publishing Australia
PO Box 194
Matraville NSW 2036
Australia
**Email** pe@pemag.au
**Web** www.electronpublishing.com

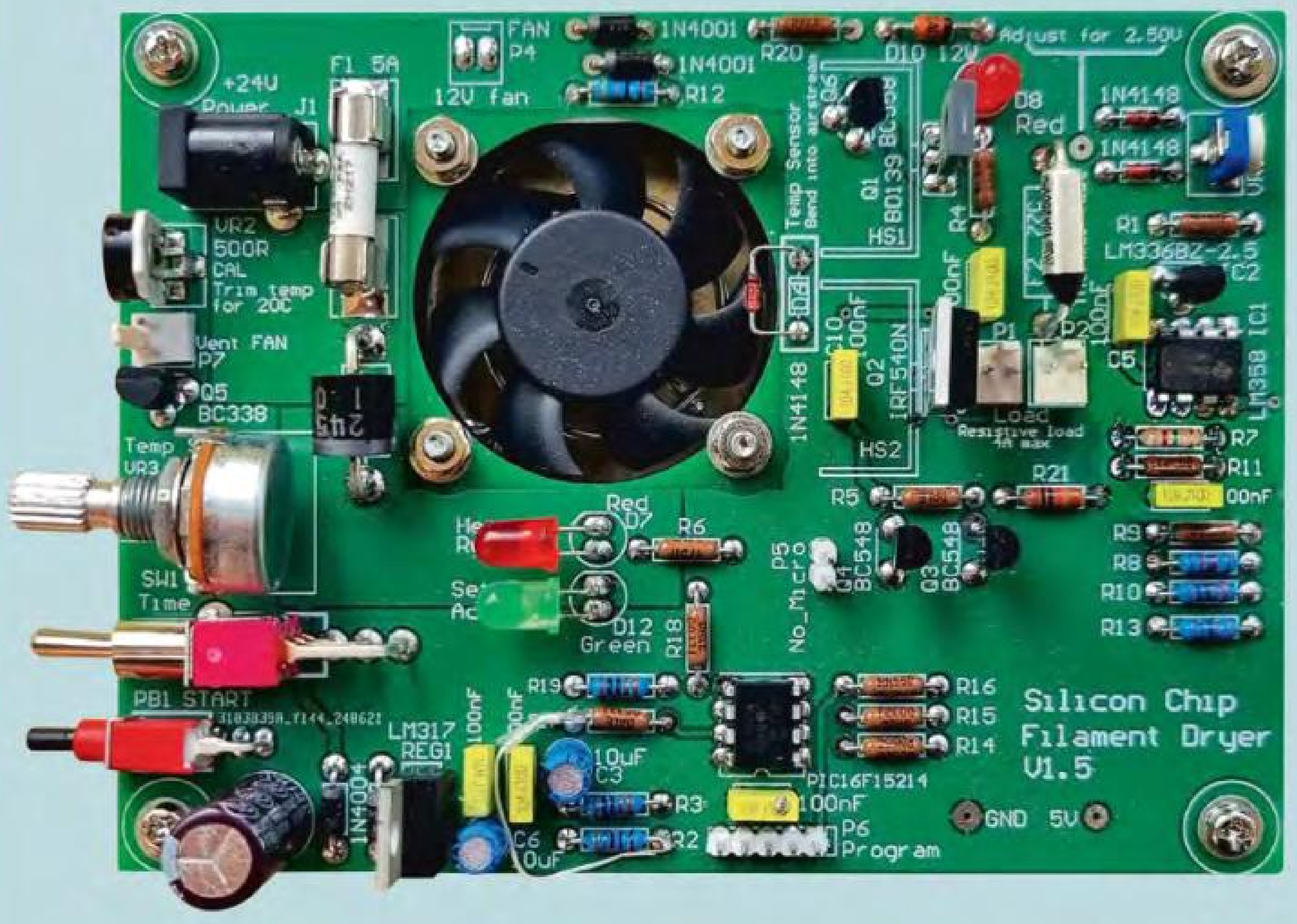
For editorial contact details, see page 2.

# Next Month – in the October issue

### Seaside Sound Simulator

Relax to the sound of electronic waves crashing on the sand.



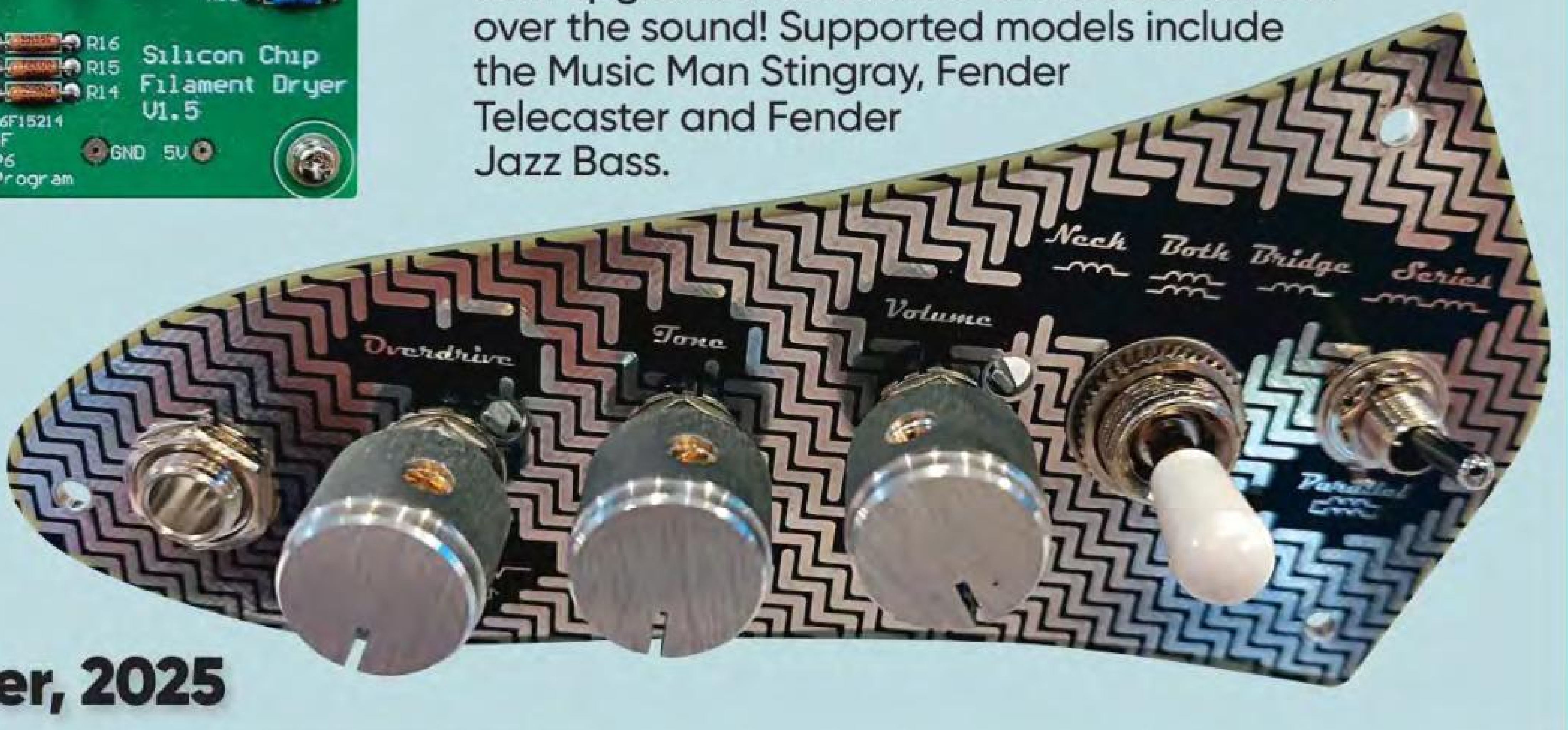


### 3D Printer Filament Dryer

Controlled heat and ventilation prevents plastic filament from absorbing moisture and ruining your prints.

### Custom electic guitar (& bass) pickguards

Jazz up one of several types of electric guitar or bass with upgraded electronics and better control over the sound! Supported models include the Music Man Stingray, Fender Telecaster and Fender Jazz Bass.



**On sale 4th of September, 2025**

Content may be subject to change

# Practical Electronics

## Subscriptions hotline

To subscribe or renew online, paying by credit/debit card or PayPal, please visit our shop at: www.electronpublishing.com

For those wishing to pay by direct debit, cheque or over the telephone, please call 'Select Publisher Services' on 01202 087631, or send a cheque (payable to: '**Select Publisher Services Ltd**') with your details to:

Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Remember, we print the date of the last issue of your current subscription in a box on the address sheet that comes with your copy.

Digital subscribers, please call 07918 614662 or visit: www.electronpublishing.com

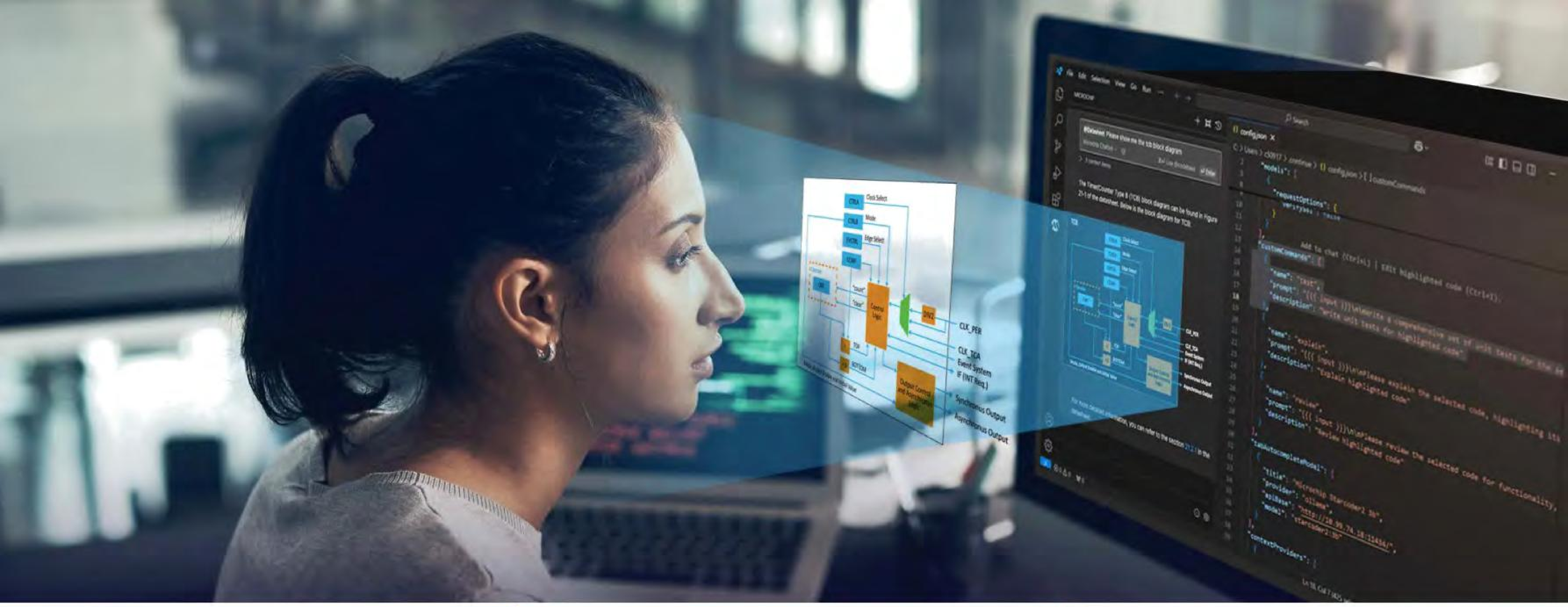
Published on approximately the first Thursday of each month by Silicon Chip Publications, PO Box 194 Matraville NSW 2036 Australia. Printed in England by The Manson Group, Reynolds House, 8 Porters Wood, Valley Road Industrial Estate, St Albans AL3 6PZ. Distributed by Seymour, 86 Newman St, London W1T 3EX. Subscriptions UK: £33.99 (6mo); £59.99 (12mo.); £114.99 (2yrs). EUROPE: airmail service, £38.99 (6mo); £69.99 (12mo); £129.99 (2yrs). REST OF THE WORLD: airmail service, £46.99 (6mo); £84.99 (12mo); £164.99 (2yrs). Make payments to 'Practical Electronics', Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH United Kingdom. Email: pesubs@selectps.com. PRACTICAL ELECTRONICS is sold subject to the conditions that it shall not, without the written consent of the Publishers first having been given, be lent, resold, hired out or otherwise disposed of by way of Trade at more than the recommended selling price on the cover, and that it shall not be lent, resold, hired out or otherwise disposed of in a mutilated condition or in any unauthorised cover by way of Trade or affixed to or as part of any publication or advertising, literary or pictorial matter.

# Did you know our online shop now sells the current issue of *PE* for £5.99 inc. p&p?

Practical Electronics
The UK's premier electronics, computing and maker magazine
Circuit Surgery
Getting an FIR filter algorithm running on a microcontroller
Audio Out
A proper PCB for building the cascode audio amplifier
Net Work
Smart meters, new Chinese EVs & Temu
Compact Hi-Fi Headphone Amp
Discrete Ideal Bridge Rectifiers
Automatic L/Q Meter
PLUS!
Techno Talk – the history of breadboarding
Cool Beans – a crash course in digital logic and gates
Dual Rail Supply Protector – a bipolar electronic fuse
Sep 2025 £5.99
www.electronpublishing.com
@practicalelec
practicalelectronics
Precision Electronics
Choosing & using voltage references
IR Remote Relay
Aug 2025 £5.99

You read that right! We now sell the current issue of your favourite electronics magazine for exactly the same price as in the High Street, but we deliver it straight to your door – and for UK addresses we pay the postage. No need to journey into town to queue outside the newsagent. Just go to our website, set up an account in 30 seconds, order your magazine and we'll do the rest.

**www.electronpublishing.com**



# Code Smarter, Faster and Better—Meet MPLAB® AI Coding Assistant

## Discover Our New, Free AI Coding Assistant Powered by Continue

The MPLAB AI Coding Assistant is here to revolutionize your coding experience. The MPLAB AI Coding Assistant is a customized version of the Continue extension from Microsoft® Visual Studio® Code (VS Code®) Marketplace, one of the leading open-source AI code assistants on the market today. This powerful tool is tailored specifically for users developing Microchip products and facilitates the development of Microchip-specific features, as well as new slash commands and shortcuts or common use cases.

Explore some of the key features that enhance your development process:

- **Pre-Configured Chatbot:** Interact with a Microchip-trained chatbot and ask Microchip-specific questions.
- **Autocomplete:** Get online code suggestions as you type to speed up development and minimize errors.
- **Search:** Quickly access Microchip datasheets directly without leaving the editor.
- **Code Generation and Review:** Generate, review, explain and comment on code effortlessly.
- **Integrated Development:** Develop and edit code without leaving the editor, with the chatbot conveniently located in the sidebar.

Users can do all this without leaving the editor, as the chatbot—once installed—sits in the sidebar. Enhance your coding experience today with our new MPLAB AI Coding Assistant, which is continuously updated with Microchip-specific information and take your development to the next level with seamless integration, real-time support and increased productivity.

MPLAB® EXTENSIONS

MICROCHIP

www.microchip.com/MPLABCodeAI

The Microchip name and logo and the Microchip logo are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
© 2025 Microchip Technology Inc. All rights reserved.
MEC2608A-UK-03-25